VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELIGENT SYSTEMS

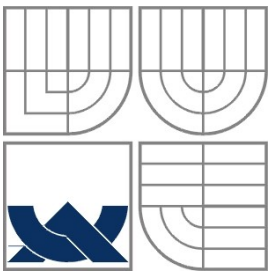
ŠIFROVÁNÍ SMS PRO MOBILNÍ KOMUNIKACI

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

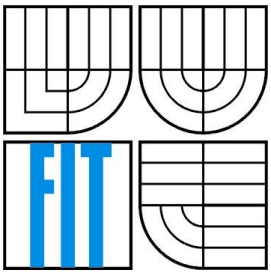
AUTOR PRÁCE
AUTHOR

BC. DAVID LISOŇĚK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELIGENT SYSTEMS

ŠIFROVÁNÍ SMS PRO MOBILNÍ KOMUNIKACI

SMS ENCRYPTION FOR MOBILE COMMUNICATION

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE BC. DAVID LISOŇĚK
AUTHOR

VEDOUCÍ PRÁCE ING. DIPL.-ING. MARTIN DRAHANSKÝ PH.D.
SUPERVISOR

BRNO 2008

Abstrakt

Tato práce se zabývá problematikou šifrování krátké textové zprávy (SMS) v mobilní komunikaci. Úvod práce je vyhrazen přehledu prvků v GSM radiotelefonních sítí a přenosu SMS zpráv v ní. Další kapitola se věnuje popisu šifrovacích metod, zejména symetrickým a asymetrickým přístupům šifrování. Návrh šifrování a dešifrování SMS zpráv v mobilní komunikaci je uveden dále. Pro šifrování a podepsání SMS zprávy je použita šifra RSA. Při šifrování se využívá schéma zarovnání OAEP. Veřejný klíč je uložen v certifikátu. Jako vhodná programovací platforma byl vybrán Symbian OS.

Klíčová slova

SMS, GSM, šifrování, asymetrická kryptografie, RSA, OAEP, certifikát, Symbian OS

Abstract

This thesis deals with encryption of short text message (SMS) in mobile communication. Introduction is dedicated to overview of the parts of GSM radiotelephone nets and SMS messages transfer . Next chapter is reserved for description of cryptographic methods especially symetric and asymetric encryptografic aproaches. A design of a SMS messages encryption and decryption techniques is in next. For SMS encryption and sign, there is used the asymetric cypher RSA. Encryption use OAEP padding schema. The public key is saved in certificate. The Symbian OS has been choosen as a suitable platform for programming of mobile devices.

Keywords

SMS, GSM, encryption, decryption, asymetric cryptography, RSA, OAEP, certificate, Symbian OS

Citace

Bc. David Lisoněk: Šifrování SMS pro mobilní komunikaci, diplomová práce, Brno, FIT VUT v Brně, 2007

Šifrování SMS pro mobilní komunikaci

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing., Dipl.-Ing. Martina Drahanského Ph.D.
Další informace mi poskytl Mgr. Petr Bouška.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Lisoněk
16.4.2008

Poděkování

Chtěl bych poděkovat mému vedoucímu Martinu Drahanskému, za rady, konzultace a upozornění na soutěž Student EEICT 2008. Také nesmím zapomenout na Petra Bouška a poděkovat mu za jeho přínosné konzultace a připomínky.

Obsah

1 Úvod

Cena informací se v dnešní době neustále zvyšuje, proto je třeba informace chránit. Při ochraně nelze spoléhat na standardní prostředky, které nejsou dostatečně silné, aby útočníkovi zamezily zneužití. Proto vznikají další ochranné prostředky, které zajišťují ochranu mnohem vyšší. Samozřejmě neexistuje naprosto spolehlivá ochrana informací, která by byla lehce využitelná v širokém měřítku aplikací. Pomocí moderních prostředků, lze ale vytvořit použitelnou ochranu, využitelnou ve většině aplikací, která bude dostatečně silná.

Hitem posledních let se staly mobilní telefony. Na začátku byly velké, těžké a téměř nepřenosné přístroje sloužící pouze k telefonování. Postupem doby, postupovala také technologie. Z velkých a těžkých se staly malé a lehké. Dnes může být takovýto telefon zaintegrován do jediného čipu. Zlepšení lze pozorovat nejen z pohledu velikosti, ale ve všech možných směrech (spotřeba energie, funkce atd). Funkce telefonu jsou dnes hlavním rozpoznávacím znakem, při jeho výběru. Dnešní telefony lze označit spíše za multimediální přehrávače, plánovače času, ale také úložiště velmi citlivých osobních či firemních informací. Hlavní funkce však zůstává stále stejná, přenos informací. Ochránit všechny tyto informace před většinou útoků, při zachování vysokého komfortu ovládání je téměř nemožné. Tato práce se zaměřuje pouze na ochranu informací posílaných v SMS (Short message service) zprávách. Práce rozebírá jak lze uplatnit moderní kryptografické metody pro zvýšení bezpečnosti námi tak ceněných informací v běžné komunikaci pomocí textových zpráv.

Cílem diplomové práce bylo navrhnout a implementovat aplikaci pro šifrování SMS zpráv. Tato diplomová práce má také za cíl posloužit jako podpora Petru Bouškovi, který se ve své disertační práci zabývá zabezpečením mobilních sítí. V první řadě bylo třeba nastudovat problematiku přenosu SMS zprávy v mobilních sítích. Struktura mobilní sítě je popsána v první části této práce. Jsou zde popsány jednotlivé komponenty GSM sítě. Je zde také probírána struktura, kódování a přenos SMS zprávy skrz GSM síť.

V další části jsou představeny moderní kryptografické algoritmy, které lze použít pro zabezpečení SMS zpráv. Cílem semestrálního projektu bylo nejen nastudovat přenos a možné způsoby šifrování SMS zpráv, ale také navrhnout vhodnou metodu výměny klíčů. Jedním z hlavních podmínek bylo, že se uživatelé nemusí nikdy osobně potkat a ani nemají zabezpečený kanál, který by bylo možné využít k výměně klíčů. Pro zabezpečení byla zvolena asymetrická kryptografie, která zajistí nejen utajení, ale také zabrání podvrhnutí SMS zprávy.

V zadání práce nebyla předepsaná žádná programovací platforma, bylo ji tedy nutno nejprve zvolit. Nabídka programovacích platforem je dnes poměrně široká. V jedné z kapitol je uveden přehled s vlastnostmi a porovnáním, jakou měrou jsou zastoupeny na trhu s mobilními telefony.

V hlavní části této diplomové práce je popsán návrh a následná implementace aplikace pro šifrování SMS. Jsou zde rozebírány důvody pro zvolení šifrovacího algoritmu RSA a programovací platformy Symbian OS. Při implementaci bylo využito projektů PyS60 a Open C, které mají za cíl rozšířit funkce a zjednodušit implementaci aplikací. V této části jsou uvedeny také implementační detaily, týkající se velikostí šifrovacího klíče, zvolení správného systému zarovnávání, popisem vytvořených tříd a stavbou aplikace s využitím PyS60 a Open C projektů. Je zde také uveden popis ovládání uživatelského rozhraní.

V předposlední části jsou popsány testy k ověření funkčnosti implementované aplikace. Testy se zabývají chybovými stavy nebo kontrolou výstupu aplikace. U chybových stavů jsou popsány situace, které vyvolají daný chybový stav, a také chybové upozornění, které je zobrazeno uživateli. Jsou zde také diskutovány možné útoky na přenos SMS zpráv. U každého útoku je popsána jeho úspěšnost na nezabezpečený a zabezpečený přenos SMS zpráv.

V závěru práce jsou uvedeny možné změny a rozšíření. Tyto vylepšení jsou navrženy pro odstranění nedostatků nalezených po testování aplikace a analýzy možných útoků. Jsou zde uvedeny, také možné rozšíření na straně uživatelského rozhraní, které by umožnily komerční využití aplikace.

2 Současný stav problematiky

Komunikace skrze SMS zprávy je v současnosti velmi využívaný komunikační prostředek. Avšak obecně se nedoporučuje zasílat v SMS zprávách důvěrné informace. Standard přenosu SMS zpráv není příliš bezpečný. Telefonní operátor dnes může číst běžně zaslané SMS zprávy. Tato vlastnost se každému nemusí zamlouvat, proto se objevil požadavek na zabezpečení SMS zpráv.

Myšlenka zabezpečit SMS zprávy není nová. Dnes lze nalézt několik firem, které pomocí kryptografických postupů zabezpečují SMS zprávy proti neoprávněnému čtení informací uložených ve zprávě.

Nejčastější koncept zabezpečení SMS zprávy je pomocí symetrického šifrovacího algoritmu AES. Šifrovací aplikace bývá většinou napsána pro platformu Java ME, ale není to podmínkou. Kryptografické zabezpečení je v tomto případě velmi silné. Největší nevýhodou této koncepce je nutnost distribuce šifrovacích klíčů. Šifrovací klíče bývají odvozeny od hesla, které si volí uživatel. Je-li heslo špatně zvoleno, je možno na něj zaútočit např. pomocí slovníku obsahující nejčastěji se vyskytující hesla. Převod hesla na šifrovací klíč probíhá nejčastěji s využitím hashovací funkce. Této koncepce využívá např. aplikace *SMS 007* firmy CircleTech s.r.o., aplikace *Fortress SMS* od firmy Silicon Village a další. Také lze nalézt bakalářské a diplomové práce, které zabezpečují tímto způsobem SMS zprávy [7][8].

Existuje také skupina aplikace, které se zabývají zabezpečením hlasové komunikace a zabezpečení SMS zpráv mají jako doplňkovou funkci. Představitelé této skupiny jsou aplikace *SecureGSM Pro* od firmy SecureGSM, *Babylon nG* od firmy Bsure Technology Ltd a další.

Pro zabezpečení lze také využít asymetrickou kryptografii. Nejčastěji je využit asymetrický šifrovací algoritmus RSA. Aplikace využívající tohoto konceptu se jmenuje *CryptoGraf Messaging v2.0* od firmy CryptoGraf. Zatím se neobjevila žádná bakalářská ani diplomová práce, která by se zabývala zabezpečením SMS zpráv pomocí asymetrické kryptografie.

Bezpečnost v mobilních telefonech lze vylepšit mnohými dalšími aplikacemi. Zabezpečit lze nejen hlasové hovory a komunikaci prostřednictvím SMS zpráv, ale také doručené emaily nebo uživatelská data, která se před uložením do paměti telefonu zašifrují. Zabezpečují se také datové přenosy. Pak se například lze připojit pomocí mobilního telefonu k zabezpečenému webovému serveru. Mnohé zabezpečovací funkce obsahují již mobilní telefony, které jsou přístupné přes API. Pro méně využívané nebo nové funkce a algoritmy bývají většinou přístupné knihovny.

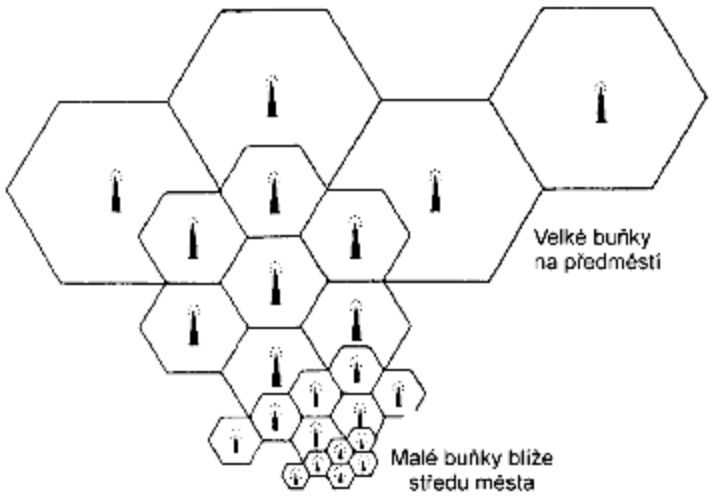
3 Přenos SMS v mobilních sítích

3.1 Struktura celulární sítě

Celulární (buňková) struktura je jeden z typů plošné struktury rádiové sítě (další jsou např. PP – point to point nebo PM point to multipoint). Tento typ struktury je využit i pro standard GSM [10]. Požadované území, například území jednoho státu, je rozděleno na velké množství malých území, buněk. Obvykle je uprostřed každé buňky umístěna základnová rádiová stanice BTS (*Base Transceiver Station*), která zajišťuje spojení mobilních účastníků v této buňce se systémem. Velikost ani rozmístění jednotlivých buněk není stejná, nejen z důvodů různého terénu, ale také podle předpokládané hustoty provozu a předpokládaného počtu účastníků.

Malé buňky s poloměrem menším než 50 metrů se nazývají pikobuňky a jsou používány pro místa s vysokou koncentrací uživatelů, např. nádraží, obchodní domy, buňky v poschodích nad sebou ve výškových budovách, atd. Mikrobuňky mají poloměr do 1 km a používají se pro oblasti s větším provozem, například v centru měst. Velké buňky nebo makrobuňky se používají pro oblasti s malou hustotou provozu, například pro pokrytí venkovských oblastí a mají poloměr až desítky km. Zvláštním typem buněk jsou buňky deštníkového typu (*umbrella cells*), které vykrývají nepokryté části území mezi menšími pikobuňkami nebo mikrobuňkami.

Několik buněk tvoří svazek buněk, jehož činnost řídí základnová řídící jednotka BSC (*Base Station Controller*), umístěná obvykle s BTS v buňce ležící uprostřed svazku. Jednotlivé BSC jsou koordinovaně řízeny z jednoho nebo několika málo center, radiotelefonních ústředen MSC (*Mobile Switching Centre*), které zajišťují spojení s jinými telekomunikačními sítěmi.



Obrázek 3.1: Část plánu městké celulární sítě [2]

Poloha mobilních stanic v síti se automaticky registruje v databázi MSC a spojení k mobilní stanici je tedy směrováno přímo do oblasti, kde se stanice nachází. Funkce umožňující najít mobilní stanici v síti, například v případě příchozího hovoru, se nazývá *roaming*. Systém monitoruje polohu mobilní stanice tím, že registruje základnovou stanici v jejímž dosahu se mobilní stanice nachází. Monitorování je prováděno v určitých časových intervalech i v době, kdy spojení se základnovou stanicí není navázáno (mobilní stanice musí být ovšem zapnuta) [1].

3.2 GSM

GSM je standard pro celulární radiotelefonní systém. Název je odvozen z názvu pracovní skupiny „Groupe Spécial Mobile“, která byla založena roku 1982. Později však bylo rozhodnuto, že se zachovají iniciály, ale změní se význam zkratky na Global System for Mobile Communications. Do češtiny se GSM překládá jako Globální Systém pro Mobilní komunikaci.
GSM patří mezi systémy druhé generace, které jsou plně digitální. Technické základy systému GSM byly definovány v roce 1987. V roce 1989 převzal kontrolu Evropský telekomunikační normalizační institut ETSI a kolem roku 1990 byla první specifika GSM na světě. V roce 1991 byla vydána první část doporučení GSM - Phase 1 [10].

Na počátku se systém používal pouze pro přenos hovorových signálů, avšak v současné době se již ve velké míře využívá také k přenosu datových signálů jako jsou SMS, obrázky atd. Je dostatečně flexibilní, aby do něj mohly být implementovány nové technologie (GPRS, HSCSD). V současné době se jeho vývoj dostal již do druhé fáze (GSM – Phase 2) a počítá se s jeho dalším vylepšováním a postupných přechodem na systém třetí generace UMTS Universal Mobile Telecommunication System.

V porovnání s analogovými systémy umožňuje dosáhnout kvalitnější spojení v nepříznivých podmínkách pozemních rádiových kanálů, efektivněji využívá přidělená kmitočtová pásma a odposlech je téměř vyloučen [1].

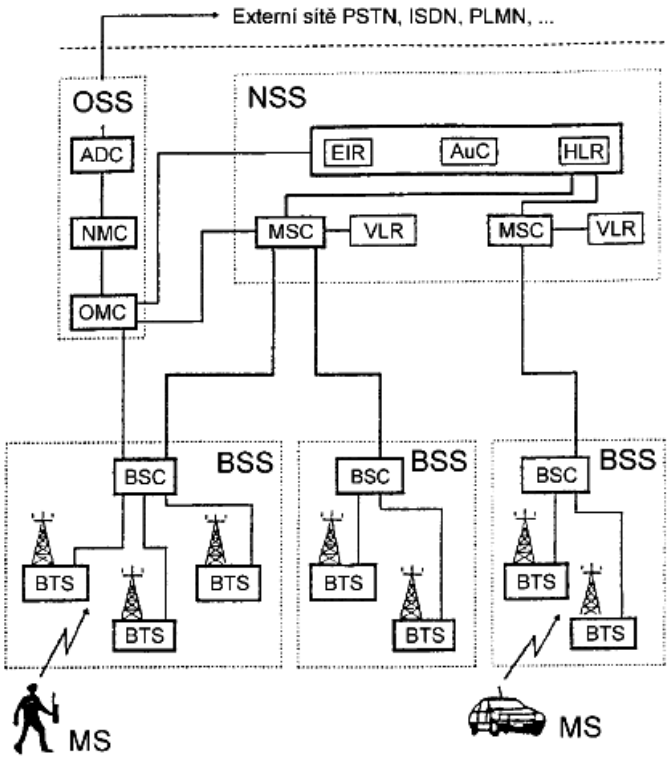
3.2.1 Architektura systému GSM

Systém GSM lze rozdělit na tyto základní části [1][11]:

- Mobilní uživatelské stanice MS (Mobile Station)
  Mobilní telefon je v podstatě transciever (vysílač/přijímač) komunikující se základnovou stanicí BTS doplněný řídícími obvody (mikroprocesor) a vstupně/výstupními zařízeními (klávesnice, display, sluchátko, mikrofon, porty). Mobilní stanice je jednoznačně identifikována pomocí čísla IMEI (International Mobile Equipment Identity), uloženého v její paměti.
- Subsystém základnových stanic BSS (*Base Station Sub-System*)
  neboli rádiový subsystém, se kterým prostřednictvím rádiového rozhraní přímo komunikují mobilní stanice MS (*Mobile Stations*).
- Síťový a spínací (přepojovací) subsystém NSS (*Network and Switching Subsystem*) označovaný někdy jako radiotelefonní ústředna s rozšířenými úkoly a funkcemi jako jsou určování polohy, handhover, přidělování kanálů apod.
- Operační subsystém OSS (*Operation Support Subsystem*)
  zajišťuje servis a koordinuje funkci celého systému - provoz, údržba, opravy poruch, atd.

Systém spolupracuje se třemi externími složkami, jsou to:

1. Uživatelé systému se svými mobilními stanicemi.
2. Operátoři, což jsou společnosti angažující se v oblasti telekomunikací, kteří řídí systém z hlediska finančního, ekonomického a částečně i provozního (účtují služby, evidence, tarifování, vydávají SIM karty, atd.).
3. Externí telekomunikační sítě, především veřejné komutované telefonní sítě PSTN (*Public Switching Telecommunication Network*), digitální sítě ISDN (*Integrated Services Digital Network*), veřejné datové sítě, atd.



BTS (*Base Transceiver Station*) základnová rádiová stanice
BSC (*Base Station Controller*) základnová řídící jednotka
MSC (*Mobile Switching Centre*) mobilní radiotelefonní ústředna
HLR (*Home Location Register*) domovský lokační registr
VLR (*Visitor Location Register*) návštěvnický lokační registr
AuC (*Authentication Centre*) centrum autentičnosti
EIR (*Equipment Identity Register*) registr mobilních stanic
IMEI (*International Mobile Equipment Identity*) mezinárodní identifikace (číslo) registrované MS
OMC (*Operational and Maintenance Centre*) provozní a servisní centrum
NMC (*Network Management Centre*) centrum managementu sítě
ADC (*Administrative Centre*) administrativní centrum

Obrázek 3.2: Architektura systému GSM [2]

BSS

1. BTS – bázový transceiver – vysílač/přijímač
2. BSC – řídící jednotka BTS

NSS

3. MSC – síť mobilních ústředen
4. HLR – domovský registr - obsahuje informace o domácích uživatelích dané sítě GSM včetně jejich aktuální pozice v rámci celé skupiny mobilních sítí sjednocených roamingovou smlouvou
5. VLR - obsahuje informace o mobilních stanicích, které se právě nacházejí v oblasti působnosti daného VLR. Tyto informace nezbytné pro realizaci služeb jsou získány z domovských registrů mobilních stanic
6. AuC - poskytuje parametry potřebné pro autentizační a šifrovací funkce pro ověření identity uživatele
7. EIR - obsahuje seznam platných mobilních terminálů definovaných pomocí IMEI (International Mobile Equipment Identity). Registr umožňuje zakázat realizovat volání z terminálů

8. SMSC (SMS centrum)
   Část sítě (ústředna), která odbavuje krátké textové zprávy v mobilní síti. Je-li mobilní telefon adresáta v okamžiku odeslání zprávy nedostupný, ukládá se zpráva po určenou dobu v SMS centru sítě.

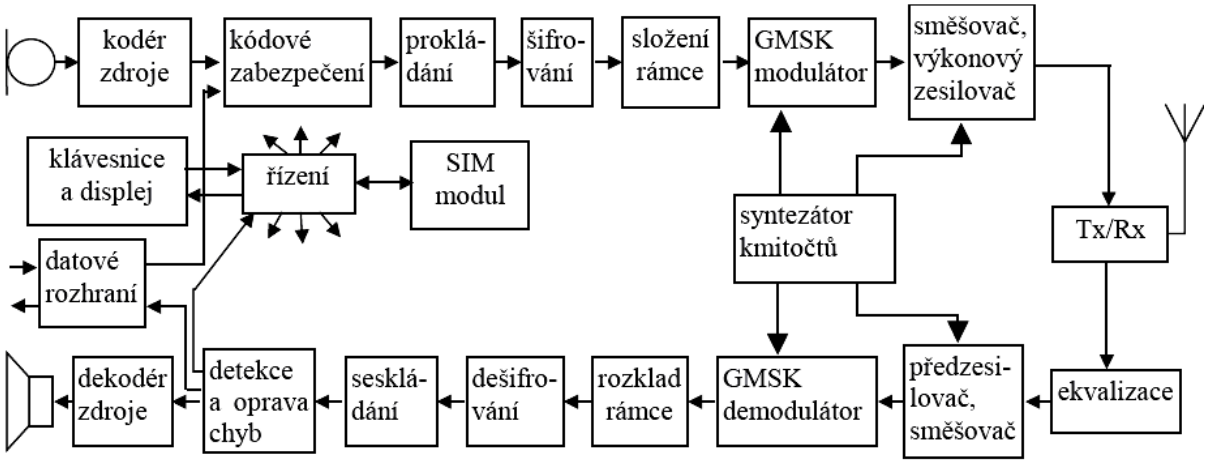
OSS

9. OMC - řídí provoz a provádí údržbu technického zázemí ostatních subsystémů sítě GSM (tj. NSS a BSS)
10. NMC - podílí se na správě mobilních stanic - tyto stanice monitoruje, zjišťuje poruchy atd.
11. ADC - podílí se na správě a managementu účastníků sítě GSM - sleduje registrace, tarifování atd.

3.2.2 Mobilní stanice systému GSM

Podle specifikací GSM se mobilní stanicí (*Mobile Station*) rozumí jednak vlastní mobilní přijímač/vysílač - mobilní telefon MT (Mobile Terminal) a jednak modul SIM – aktivační prvek pro GSM - ten umožňuje unikátní identifikaci uživatele v rámci celé sítě GSM. Mobilní stanice obsahuje vysílač/přijímač na pásmu 900 (1800, 1900) MHz s digitálním způsobem přenosu umožňující komunikaci se základnovými stanicemi. Je tvořen GMSK modulátorem/demodulátorem, směšovači up/down, a zesilovacími stupni vysílací/ přijímací. Dále obsahuje audiokodek s mikrofonem a se sluchátkem a rozhraní pro kontakt s uživatelem - klávesnici, displej, jakož i další rozhraní pro doplňkové služby - konektor pro připojení data/faxové karty nebo přímo sériové rozhraní (je-li modem součástí telefonu), **SMS** vysílač/přijímač (typicky tvořený kombinací předešle uvedených částí).

Mobilní stanici lze používat jen vložením aktivní SIM karty (výjimkou jsou tísňová volání 112). SIM karta obsahuje základní informace o majiteli - jeho identifikační kód, telefonní seznamy, informace o předplacených službách apod. Její zneužití je blokováno čtyřmístným PIN kódem, jenž si může každý libovolně nastavit. Při jeho opakovaném trojnásobném chybném zadání se SIM karta zablokuje a je potřeba ji odheslovat zadáním zvláštního kódu PUK.



Obrázek 3.3: Blokové schéma mobilní stanice [2]

3.2.3 SMS v GSM sítích

Výměny krátkých textových zpráv je mezi uživateli sítě GSM velmi oblíbenou službou. Zkratka SMS pochází z anglického názvu Short Message Service. Tato služba byla definována jako část standardu GSM v roce 1985 (dokumenty GSM 03.40 a GSM 03.38). [15]

Každá SMS zpráva se skládá z několika částí. První část je uživateli skryta a obsahuje informace o správném doručení zprávy jako je číslo SMS centra, popis protokolu, způsob kódování dat, doba platnosti zprávy, délku uživatelského textu. Tyto informace jsou ke zprávě přidány automaticky zařízením, které zprávu posílá. Další částí je uživatelský text. Tento text je omezen 140 byty (8-bitů) textu.

- Prostor pro uživatelskou informaci: 140 * 8 bitů = 1120 bitů

Do tohoto prostoru lze vložit 160 7-bitových znaků. Znaky byly vybírány s ohledem na jejich maximální frekventovanost v textové komunikaci nejedná se tedy např. o 7-bit ACSII kódování. Seznam všech 7-bitových znaků je uveden v tabulce 1.

- Počet 7-bitových znaků: 1120 : 7 bit = 160

Tabulka 1: znaky použité v SMS [8]

GSM 03.38																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	@	£	$	¥	è	é	ù	ì	ò	Ç	LF	Ø	ø	CR	Å	å
1x	Δ	_	Φ	Γ	Λ	Ω	Π	Ψ	Σ	Θ	Ξ	ESC	Æ	æ	ß	É
2x	SP	!	"	#	¤	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	¡	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ñ	Ü	§
6x	¿	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ñ	ü	à

Toto zkrácené kódovaní znaků však nepokrývá všechny symboly a informace. Při nutnosti posílání 8-bitových informací, tedy klasických bytových dat jako jsou např. vyzváněcí melodie, loga na displej mobilního telefonu nebo konfigurační SMS (např. konfigurace WAPu), je prostor 140 bytů jak bylo uvedeno výše.

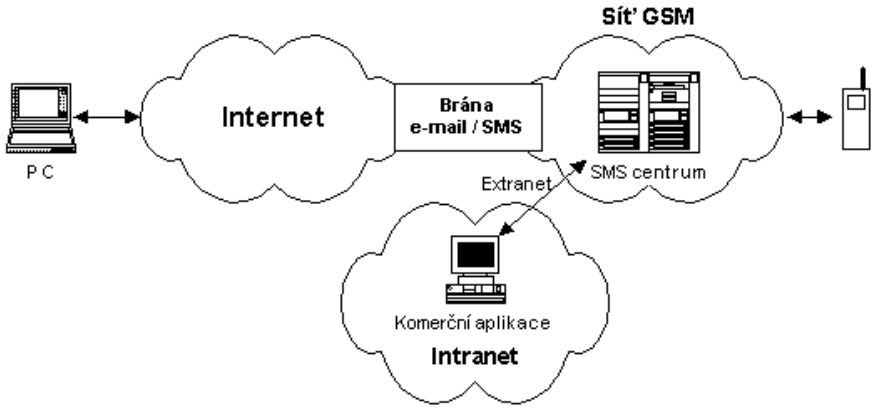
Existují také informace, které musí být kódovány 16-bitově. Zde patří kódování znaků národních abeced v SMS zprávách, pomocí kódování Unicode (UCS-2). Toto se týká také češtiny u uživatelů tolik neoblíbeného používání českého slovníku T9 s diakritikou, z důvodů zkrácení zprávy. Šestnácti bity se kódují také Flash SMS zprávy, které se po přijetí automaticky zobrazí na displeji mobilního telefonu [15][16].

- Počet 16 bitových znaků: 1120 : 16 bitů = 70

Při překročení délky textu se SMS zpráva rozdělí do více zpráv, které se odešlou odděleně. Dnešní mobilní telefony dokáží takovéto zprávy identifikovat a zobrazit je uživateli jako jednu dlouhou celistvou zprávu. Takováto dlouhá SMS zpráva může být složena z 6-8 klasických SMS zpráv.

Po odeslání zpráva putuje do SMS centra. Číslo SMS centra bývá obvykle uloženo na SIM kartě. Po doručení do SMS centra, je zpráva uložena. Následuje pokus o doručení zprávy adresátovi. Pokud není adresát připojen k síti, nebo doručení selže, pokusí se SMS centrum doručit zprávu později. Pokusy o doručení trvají dokud zprávě nevyprší platnost.

Adresát nemusí být určen pouze číslem telefonu v GSM sítí. SMS zprávy lze zasílat také jako email, fax, existuje dokonce služba, která uživateli pevné linky SMS zprávu přečte elektronickým hlasem. O převod, takovýchto SMS zpráv se stará právě SMS centrum. Nestandardní nemusí být pouze adresát. SMS zprávy lze zasílat také prostřednictvím internetových SMS bran. Mnozí operátoři tuto službu poskytují zdarma pro nekomerční využití. SMS zpráva může být zaslána také jako email, na speciální emailovou adresu, která je spojena s číslem mobilního telefonu v GSM sítí. Tuto službu si však musí uživatel zpřístupnit u svého operátora. U všech těchto služeb stále platí velikostní omezení SMS zprávy, proto jsou zprávy ořezávány nebo rozděleny do více SMS zpráv.



Obrázek 3.4: schéma různých cest přenosu SMS zpráv [10]

Užitečnou vlastností SMS zpráv je možnost opatřit je tzv. notifikací přijetí zprávy. Tedy označení zprávy, tak aby ji GSM síť sledovala a oznámila odesilateli, zda byla zpráva příjemci doručena.

Důležitou vlastností přenosu SMS zpráv je to, že odesílání nebo příjímání SMS zpráv může být realizováno nezávisle na hlasové/datové komunikaci uživatele mobilního telefonu. Pro přenos SMS zpráv se nesestavuje virtuální okruh, který se vytváří pro navázání hovoru. Tedy uživatel může současně telefonovat a přenášet SMS zprávy [17].

4 Šifrování

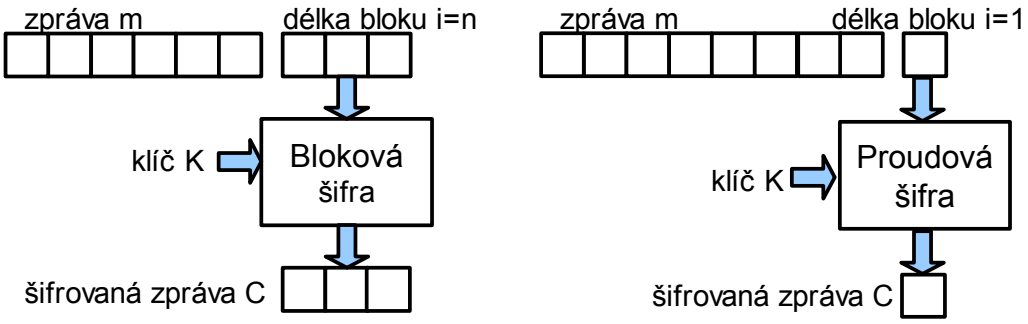
První známky o utajování informací máme již ze starého Egypta, kde pro utajení textu použili speciální hieroglyfy. Další známou starověkou šifrou je Caesarova šifra [12], kterou používal pro vojenskou komunikaci Julius Caesar. Myšlenka této šifry je taková, že jsou jednotlivá písmena v abecedě ve zprávě posunuta o tři písmena vpřed. Postupně se šifry zdokonalovaly. V 19. století se začaly rozvíjet šifrovací stroje, které nahradily šifranty a umožnily rychlejší a důmyslnější šifrování. Další významné urychlení a rozmach přišel s rozvojem výpočetní techniky. V současné době je šifrování široce používáno nejen k vojenským a politickým účelům, ale je využíváno i širokou věřeností, i když o tom třeba ani neví (telefonní hovory, televizní vysílání, přístup na webové stránky atd.).

Věda zabývající se šifrováním se nazývá kryptografie. Kryptografie využívá matematických poznatků nejen k utajení - *důvěrnosti* informace, ale také pro zaručení *integrity, autentizace a nepopiratelnosti.*

- *důvěrnost* – ochrana proti neautorizovanému zpřístupnění důvěrné informace
- *integrita* – ochrana proti neautorizované změně dat
- *autentizace* – prokázání totožnosti subjektu
- *nepopiratelnost* – zaručené prokázání původu zprávy

Naopak kryptoanalýza je věda zabývající se získání důvěryhodných informací ze zašifrovaných dat bez znalosti klíče. Díky kryptoanalýze máme prostředky pro ověření bezpečnosti kryptografických mechanizmů. Jeden z požadavků říká: šifrovaný text by měl mít stejné statistické vlastnosti jako text náhodný. Kryptologie se zabývá jak kryptologií tak kryptoanalýzou. [3]

Pokud se při šifrování i dešifrování používá stejný klíč jedná se o symetrickou kryptografii. Je-li klíč pro šifrování odlišný od klíče pro dešifrování hovoříme o asymetrické kryptografii. Šifrovací algoritmy můžeme také dělit na základě kriterií práce s daty na šifry *proudové*, *blokové*.



Obrázek 4.1: Bloková a proudová šifra

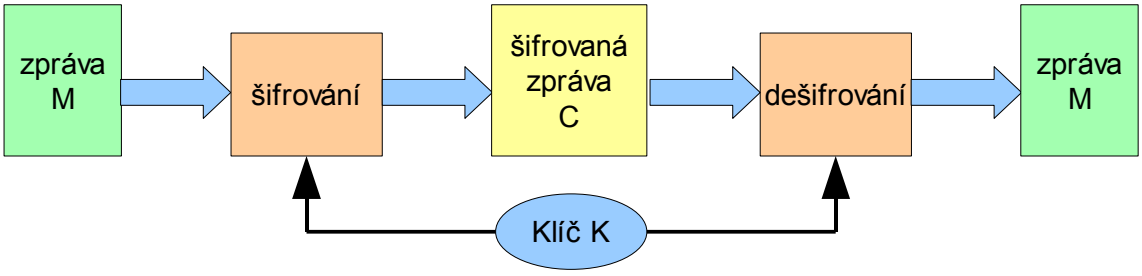
Proudová šifra bere zprávu po bytech (případně po bitech). Každý byte zašifruje a předá dále. Hlavní výhodou je, že celá zpráva nemusí být k dispozici na začátku šifrování. Toho lze využít zejména v telekomunikacích, kde lze šifrovat datové toky. Proudové šifry bývají lépe použitelné, ale jsou obecně méně bezpečné. Proudové šifry můžeme rozdělit na synchronní a samosynchronizující. U synchronní proudové šifry je pseudonáhodná posloupnost (key stream) nezávislá na zprávě. Šifrovaná zpráva bývá obvykle počítána jako XOR otevřené zprávy a key streamu. U samosynchronizující proudové šifry je šifrovaná zpráva spočtena z otevřené zprávy, key stremu a *n* předcházejících bytů šifrované zprávy. Při chybě v šifrované zprávě se mechanizmus synchronizuje po načtení *n* správných bytů šifrované zprávy.

Bloková šifra bere zprávu po blocích větší velikosti než byte. Obvykle bývá velikost bloku shodná s velikostí klíče (nejčastější hodnoty bývají 64, 128, 256 bitů u symetrických šifer). Při použití blokového přístupu však musíme ošetřit případ kdy zpráva má menší velikost než bloku. Tento problém řeší zarovnávání (Padding). Před šifrováním musí být k dispozici celý blok zprávy, proto tento typ šifry není příliš vhodný pro šifrování proudů dat. Výhodou je, že díky blokovému přístupu jsou blokové šifry bezpečnější vůči slovníkovým a statistickým útokům. Většina dnes používaných algoritmů jsou blokové šifry.

4.1 Symetrická kryptografie

Symetrické šifrovací algoritmy používají k šifrování i dešifrování dat stejný klíč. Obě strany tedy musí sdílet tajemství (klíč). Tato vlastnost je však velkou nevýhodou. Před začátkem komunikace je nejprve nutno zvolit tajný šifrovací klíč. Klíč však nesmí být nikým během výměny odposlechnut. Naopak výhodou symetrických šifrovacích algoritmů bývá jejich vysoká rychlost oproti nesymetrickým kryptovacím algoritmům. V praxi jsou používány zejména pro šifrování velkých objemů dat a proudových dat. Moderní symetrické šifry jsou šifry blokové. Vychází obvykle z Feistlovy šifry, což je substitučně permutační síť.[2][13]

- šifrování zprávy *m* klíčem *k*: *c=e(m,k)*
- dešifrování zprávy *m* klíčem *k*: m=*d(c,k)*



Obrázek 4.2: symetrická šifra

4.1.1 DES

Algoritmus DES (Data Encryption Standard) je jeden z nejpoužívanějších šifer. Vychází z algoritmu Lucifer, který byl vyvíjen v sedmdesátých letech firmou IBM. V roce 1976 vyhrála soutěž šifer vypsanou americkým úřadem pro bezpečnost. Požadavky na algoritmus byly: vysoká bezpečnost, přesná specifikace, otevřenost algoritmu, lehká realizovatelnost pomocí hardwaru a rychlost. Délka šifrovacího klíče je 64 bitů, z toho je 56 bitů použito k šifrování a zbylých 8 bitů je parita. Šifra je bloková. Velikost bloku je roven velikosti klíče 64 bitů. Šifrování je realizováno v 16 kolech. Pro každé kolo je vytvořen jeden 48 bitový subklíč. Nad vstupními daty se provede počáteční permutace. Následuje 16 kol. V každém kole se vymění prvních 32 bitů za druhých 32 bitů šifrovaných dat. V pravé polovině proběhne expanze z 32 bitů na 48 bitů se kterými je proveden XOR se subklíčem daného kola. Následuje substituce pomocí S boxu, která provede i redukci ze 48 bitů na 32 bitů. Pak je provedena permutace a XOR s levou polovinou.

Hlavní slabiny algoritmu DES jsou malá délka klíče (snaha o zlepšení TRIPLEDES), možnost zadních vrátek v S boxech, existence slabých klíčů.

4.1.2 AES

Algoritmus AES (Advanced Encryption Standard) je nástupcem algoritmu DES. Byl standardizován americkou vládou v roce 2001. Je založen na algoritmu Rijndael od autorů Joan Daemen a Vincent Rijmen z Belgie. Jedná se o blokovou šifru o velikosti bloku 128bitů. Délky klíčů jsou 128, 192 nebo 256 bitů.

AES pracuje v kolech. Počet kol se liší podle velikosti klíče.V prvním kroku šifrování se provede operace *Add Round Key* (XOR subklíče s blokem). Následuje provedení kol. Každé kolo obsahuje operace *Byte Sub* (nahrazení každého bytu v bloku podle S-boxu), *Shift Row* (byty jsou uspořádány do matice a posunuty), *Mix Column* (násobení matic) a operaci *Add round key*.

4.1.3 IDEA

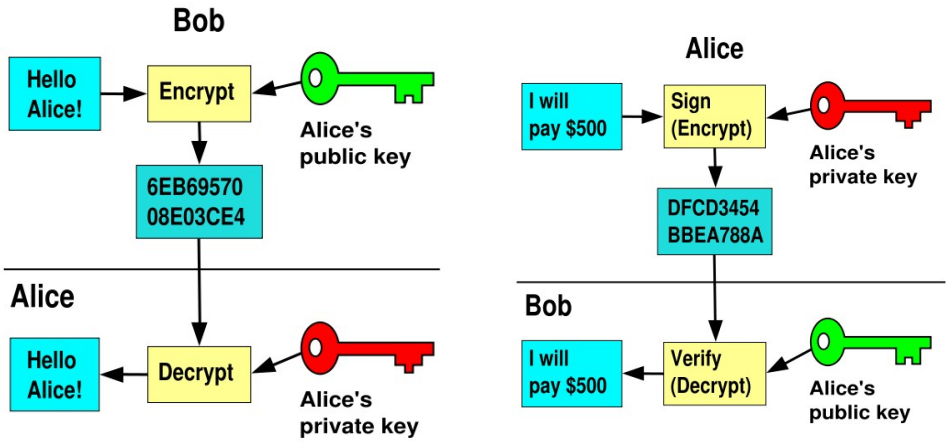
Algoritmus IDEA (International Data Encryption Algorithm) byl uveden na začátku devadesátých let jako náhrada nedostačujícího algoritmu DES. K masivnímu rozšíření však brání patent, šifra může být použita zdarma jen k nekomerčním účelům. Jedná se o blokovou šifru, která pracuje s velikostí bloku 64 bitů. Velikost klíče je 128 bitů. Šifrování probíhá v 8 kolech. Každé kolo obsahuje operace XOR, modulární sčítaní a modulární násobení. Správným uspořádáním těchto operací se dociluje dobré bezpečnosti.

4.2 Asymetrická kryptografie

Asymetrické šifrovací algoritmy nazývány také jako šifry s veřejným klíčem byly objeveny později než symetrické šifrovací algoritmy. Základní myšlenka je ta, že se místo jednoho tajného klíče použijí klíče dva. Jeden soukromý (privátní) klíč který se tají. Druhý veřejný který je dán k dispozici všem bez omezení. K šifrování zprávy odesilatel použije veřejný klíč příjemce. K dešifrování příjemce použije svůj soukromý klíč. Tímto odpadá nutnost výměny tajného šifrovacího klíče. Veřejný a soukromý klíč musí být spolu matematický svázány, ale nesmí být možné z veřejného klíče nijak odvodit klíč privátní[2][14].

- šifrování zprávy *m* veřejným klíčem k_v: $c = e(m, k_v)$
- dešifrování zprávy *m* soukromým klíčem k_s: $m = d(c, k_s)$

Asymetrické kryptografie nemusí sloužit pouze k *důvěrnosti* (utajení) dat, lze ji využít také pro zajištění *autenticity*, *integrity* a *nepopiratelnosti*. Pro zajištění *nepopiratelnosti*, kterou samotné šifrování neposkytuje, slouží elektronický podpis. Typické použití digitálního podpisu bývá následující: Odesilatel vytvoří hash zprávy. Hash má malou velikost a tedy podepsání - zašifrování soukromým klíčem odesilatele je operace výpočetně přijatelná oproti nutnosti šifrovat velkou zprávu „pomalou“ asymetrickou šifrou. Zpráva je digitálně podepsaná, když je zašifrovaná soukromým klíčem. *Autenticita* a *nepopiratelnost* je zaručena znalostí soukromého klíče pouze odesilatelem zprávy, nikdo jiný tedy nemůže být autorem. Odesilatel nemůže popřít autorství dokumentu. *Integrita* zprávy se ověří vypočtením hashe doručené zprávy a porovnání s hashem z digitálního podpisu. Hash z digitálního podpisu se získá dešifrováním pomocí veřejného klíče odesilatele. Odpovídají si oba hashe zpráva nebyla během přenosu modifikována, nebyla tedy porušena její *integrita*.



Obrázek 4.3: Asymetrická kryptografie - šifrování, digitální podpis [7]

4.2.1 Šifra RSA

Algoritmus RSA, který byl objeven Ronem Rivestem, Adim Shamirem a Leonardem Adelmanem v roce 1978. Jedná se o asymetrický šifrovací algoritmus založený na faktorizaci velkých čísel. RSA je šifrovací algoritmus vhodný jak k šifrování tak podepisování. Jedná se o blokovou šifru.

Před vlastním šifrováním je třeba vygenerovat soukromý a veřejný klíč. Generování klíčů je typicky výpočetně náročná operace. Velikost klíče bývá typicky 512, 1024, 2048 nebo 4096 bitů. Generování probíhá podle následujícího postupu.

1. vygeneruj dvě náhodná velká prvočísla p a q
2. spočti součin $n = p \,.\, q$
3. spočti hodnotu Eulerovy funkce $\varphi(n) = (p - 1) \,.\, (q - 1)$
4. zvol hodnotu e, takovou že $e < \varphi(n)$ a e je nesoudělné s $\varphi(n)$
5. spočti d tak, že $d = e^{-1} \bmod \varphi(n)$

Veřejný klíč je dvojice (n, e), soukromý klíč je dvojice (n, d). Hodnota n se označuje jako veřejný modulus. Hodnota e se označuje jako veřejný exponent, typicky nabývá lichých hodnot 3, 17 nebo 65537. Hodnota d se označuje jako soukromý nebo dešifrovací exponent.

Šifrování a dešifrování jsou velmi podobné operace, proto je algoritmus RSA vhodný pro šifrování i podepisování. Zpráva kterou chceme zašifrovat je brána jako velké celé číslo m. Při šifrování veřejným klíčem se utajená zpráva c se spočte podle vzorce $c = m^e \bmod n$. Dešifrování soukromým klíčem se provádí následovně $m = c^d \bmod n$.

Bezpečnost algoritmu závisí na obtížnosti faktorizace velkých čísel. Slovem faktorizace se v matematice označuje problém rozložení čísla na součin menších čísel. Nejčastěji se rozkládá celé číslo na součin prvočísel. Pro tuto úlohu však dosud není znám žádný efektivní algoritmus. Při šifrování je velmi vhodné použít správný systém zarovnávání. Je-li algoritmus RSA použit bez správného systému zarovnávání může trpět následujícími problémy.

- je-li hodnota zprávy m taková, že $m = 0$ nebo $m = 1$, pak je vždy šifrovaná zpráva $c = 0$ nebo $c = 1$.
- je-li použit malý veřejný exponent např. $e = 3$ a malá hodnota m, nemusí se uplatnit modulární operace s n. Pak může být zpráva snadno dešifrována s použitím veřejného exponentu bez nutnosti znalosti modulu.
- Algoritmus RSA je deterministický, proto je náchylný ke slovníkovým útokům.

S těmito problémy se zejména setkáme při šifrování malých zpráv, složených z ASCII znaků. Pro odstranění je vhodné použít systém zarovnávání s nedeterministickou složkou, který také zabezpečuje že hodnota *m* nebude nabývat kryptograficky slabých hodnot.

4.2.2 Šifra Diffie-Hellman

Algoritmus Diffie-Hellman je první algoritmus s veřejným klíčem, založený na problému diskrétních logaritmů. Autoři Whitfield Diffie a Martin Hellman publikovali algoritmus v roce 1976. Algoritmus je vhodný pro ustanovení klíče relace *K*. Klíč relace *K* se použije pro další šifrování komunikace symetrickou šifrou. Algoritmus je díky své jednoduchosti velmi rychlý. Zvolení čísla relace probíhá následovně:

1. zvolí se velké prvočíslo *n* a hodnota *g*, která nedělí *n*
2. první účastník si zvolí hodnotu *x*, ze které spočte $X = g^x \ mod\ n,$ hodnota *X* je zaslána druhému účastníkovi
3. druhý účastník si zvolí hodnotu *y*. Ze zvolené hodnoty y a doručené hodnoty *X* si spočte klíč relace $K = X^y \ mod\ n$.
4. Prvnímu účastníku je zaslána zpráva $Y = g^y \ mod\ n$
5. Ze zprávy *Y* si první účastník spočte klíč $K = Y^x \ mod\ n$

4.2.3 šifra DSS

DSA (Digital Signature Algorithm) je algoritmus pouze pro digitální podpis, nedá se tedy použít pro šifrování ani distribuci klíčů. Americká vláda ho přijala jako standard digitálního podpisu DSS (Digital Signature Standard) v roce 1993. Algoritmus je založen na problému diskrétního algoritmu. Velikost klíče byla původně 512 bitů, později byla však zvětšena na 1024 bitů. V porovnání s RSA je rychlejší při podpisu, ale pomalejší při jeho ověření. V bezpečnosti je algoritmus DSA porovnatelný s algoritmem RSA. Nevýhoda je, že pro některé specifické hodnoty je problém diskrétního logaritmu snadno řešitelný. Také panují obavy, že v parametrech standardizovaného algoritmu mohou být zadní vrátka.

4.2.4 ElGamal

Algoritmus ELGamal je algoritmus vhodný jak pro šifrování, tak pro digitální podpis. Výpočet šifrování je však velmi odlišný od výpočtu digitálního podpisu. Algoritmus je založen na problému diskrétního logaritmu a vychází z algoritmu Diffie-Hellman. Název algoritmu je odvozen od jména autora Tahela Elgamala. Největší nevýhodou je skutečnost, že šifrovaná zpráva je dvakrát delší než

otevřená zpráva. V porovnání s algoritmem RSA je ELGamal výpočetně náročnější. Pro generování soukromého a veřejného klíče slouží pouze tři hodnoty. Prvočíslo *p,* číslo *q* obě co nejvyššího řádu a náhodná hodnota a, která je menší než p. Z těchto hodnot se spočte hodnota r $= q^a \ mod \ p$. Veřejný klíč je pak složen z trojice (*p, q, r*). Soukromý klíč je pouze číslo *a* [12].

4.2.5 Eliptické křivky

Eliptické křivky EC (Elliptic Curve) jsou předmětem zkoumání déle než 150 let. Nasazení v kryptografii se zkoumá však až posledních 20 let. Zájem byl hlavně z důvodu hledání alternativy k patentovaným algoritmům. Algoritmy založené na Eliptických křivkách nejsou dnes ještě dostatečně široce rozšířeny, mají však velký potenciál. Rozšíření hlavně brání slabá podpora v šifrovacích knihovnách. Pro šifrování lze využít problému diskrétního logaritmu převedeného do prostředí eliptických křivek tzv. ECDLP (Elliptic Curve Discrete Logarithm Problem). Hlavními výhodami eliptických křivek jsou:

- Kryptosystémy s eliptickými křivkami poskytují největší míru obtížnosti na jeden bit délky klíče ze všech známých asymetrických systémů.
- Zdá se, že problém ECDLP je mnohem obtížnější než problémy faktorizace a diskrétního logaritmu.
- Síla kryptosystému s eliptickými křivkami stoupá s délkou klíče rychleji než síla RSA.
- Výpočetní složitost u eliptických křivek je menší než u faktorizace nebo diskrétního logaritmu [2].

4.3 Zarovnávací schémata

Při použití blokové šifry se nevyhneme problému rozdílné velikosti bloku pro šifrování a velikosti zprávy. V případě, že je zpráva menší nebo po rozdělení zprávy na bloky zbude část, která je menší než blok, musíme zprávu doplnit do plné velikosti bloku. Pro doplnění zprávy se používají zarovnávací schémata. Zarovnávací schéma musí ke zprávě připojit vyplňkovou informaci, kterou lze následně odebrat a přitom nepozměnit původní zprávu. Vyplňovací schéma by také nemělo negativně ovlivnit bezpečnost šifrovacího algoritmu [2].

V asymetrické kryptografii se ujala sada standardů Public Key Cryptography Standards (PKCS). Jedná se o průmyslové standardy, které původně navrhla společnost RSA Security. RSA Security, výzkumná divize RSA Labs, je zainteresovaná v propagaci a snaží se usnadnit použití asymetrické kryptografie v praxi.

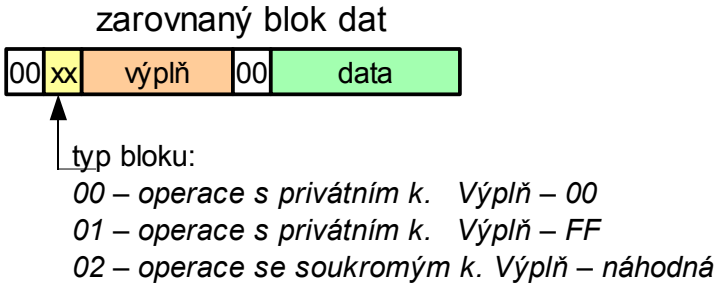
4.3.1 Jednoduché zarovnávání

Nejjednodušším typem je vyplnit zprávu do velikosti bloku hodnotami 0, to jedná se však o nejednoznačné zarovnání. Příjemce neví kolik hodnot má odstranit. V praxi se proto používá jednoznačné zarovnání. Za zprávu se zapíše hodnota 1 a zbytek bitu se dolní hodnotou 0. Toto schéma však není vhodné pro asymetrickou kryptografii z důvodu slabé bezpečnosti.

4.3.2 Zarovnávání podle PKCS1 v1.5

Podle standardu PKCS1 verze 1.5 se pro vyplnění využívá také náhodné informace. Nejčastější použití je v protokolu SSL/TSL, které využívají webové servery a prohlížeče. Na tento princip byl proveden Bleichenbacherův útok. Na toto zarovnávací schéma neexistuje formální analýza bezpečnosti. Formát vyplněného bloku je následující [9]:

1. Počáteční byty mají hodnoty 00. To zaručuje, že šifrovaný blok, převedený na integer, je menší než modulus.
2. Další dva byty určují *typ bloku*. Hodnota 00 nebo 01 značí operace s soukromým klíčem, hodnota 02 pak značí operaci s veřejným klíčem.
3. Vyplňovací řetězec bytů. V případě, že je *typ bloku* 00, řetězec obsahuje hodnoty 00. Pro *blok typu* 01, obsahuje řetězec hodnoty FF. Pro *blok typu* 02 jsou generovány pseudonáhodné hodnoty bez hodnoty 00.
4. Následuje oddělovací hodnota 00. Pro *blok typu* 00 musí data začínat nenulovou hodnotou, nebo musí být známa jejich délka. Pro *typy bloků* 01 a 02 je oddělovací hodnota také 00.
5. Data jejichž velikost nedosahovala plné velikosti bloku.

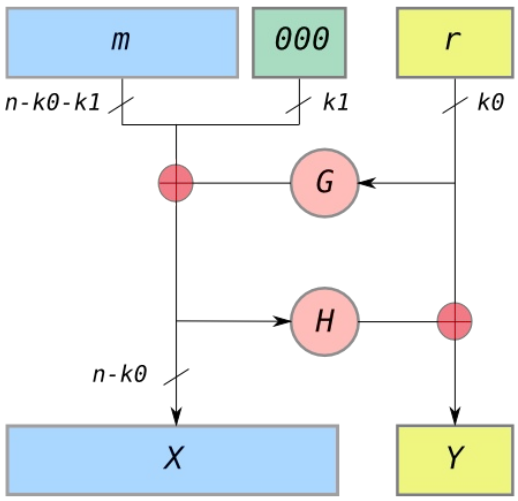


Obrázek 4.4: Zarovnání podle PKCS1 v1.5

4.3.3 OAEP

Zarovnání OAEP (Optimal Asymmetric Encryption Padding) je definováno ve PKCS1 verze 2.1. Jedná se o nástupce PKCS1 verze 1.5. OAEP je CSS (chosen cipher text security) pokud jsou použita „náhodná orákula“ pro funkce H a G. V praxi se pro funkce H a G používají hashovací funkce. Při implementaci je třeba dávat pozor na postranní kanály, kdy je možno dát útočníku užitečnou informaci např. z doby zpracování zarovnání ve fyzickém zařízení. Zarovnání do bloku probíhá následovně [24]:

1. Číslo *n* je počet bitů modulusu u RSA šifry.
2. Vygeneruje se náhodné číslo *r* o velikosti *k0 bitů.*
3. Za zprávu *m* se doplní *k1* bitu s hodnotou 00, aby celá velikost bloku byla rovna veřejnému modulusu u šifry RSA o velikosti *n*. Tedy zpráva má délku $n - k0 - k1$.
4. Náhodné číslo je expandováno maskovací funkcí *G* na délku doplněné zprávy o velikosti $n - k0$.
5. Je proveden XOR doplněné zprávy a výstupu funkce *G*.
6. Tento výstup je uložen do výstupního bloku *X* a také použit jako vstup maskovací funkce *H*, jejíž výstup má velikost délky náhodného čísla *k0.*
7. Je proveden XOR výstupu funkce *H* a náhodného čísla, výsledek je přidán do výstupního bloku *Y* o velikosti *k0.*
8. Výstupní blok, vzniklý spojením bloků *X* a *Y* má správnou velikost *n* a je možno ho zašifrovat.



Obrázek 4.5: Zarovnávání OAEP [23]

4.4 Asymetrická správa klíčů

Může se zdát, že asymetrickou kryptografií jsme vyřešili problém distribuce klíče nezabezpečenou cestou. Útočník může znát jak veřejný klíč příjemce tak zašifrovanou zprávu, přesto nebude schopen zprávu dešifrovat. Nesmíme však zapomenou na útok, kdy je podvržen veřejný klíč. Tomuto útoku se snaží zabránit systém certifikátů. Certifikát je datová struktura svazující identitu držitele s jeho veřejným klíčem. Pro ověření toho spojení slouží certifikační autorita, která certifikát také vydává pro držitele soukromého a veřejného klíče. Certifikační autorita pak ručí za správnost údajů uložených v certifikátu. Certifikační autorita je subjekt, kterému všichni účastnící důvěřují. [2]

4.4.1 Certifikát

Certifikát poskytuje vazbu mezi identitou a veřejným klíčem vlastníka. Je vydává v určité třídě ověření. Certifikát obsahuje:

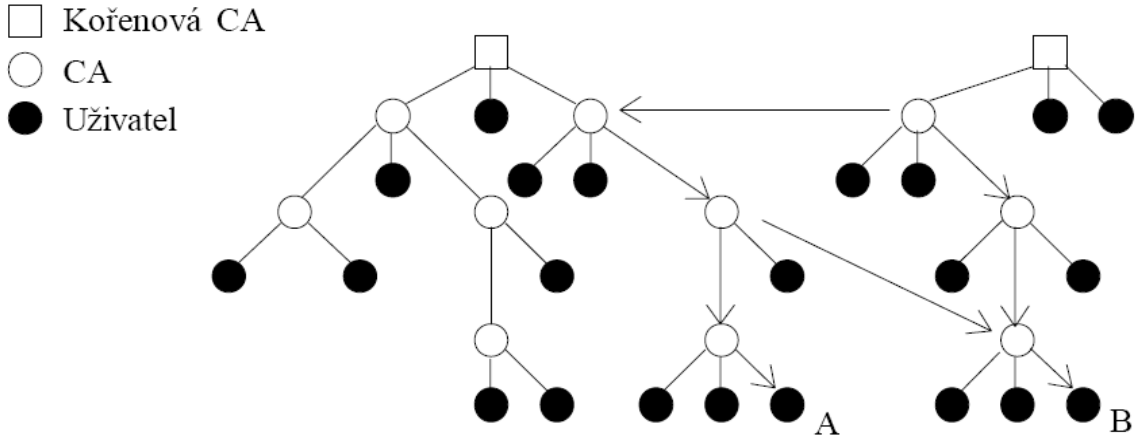
- jméno vlastníka
- jednoznačné sériové číslo certifikátu v rámci certifikační autority
- platnost certifikátu vymezena daty od, do
- vydavatele – identifikaci certifikační autority
- veřejný klíč vlastníka
- identifikaci algoritmu pro který lze veřejný klíč použít
- verze formátu certifikátu
- podpis certifikační autority
- a další

Dnes nejpoužívanější formát pro uložení certifikátů je podle normy X.509. Do souboru se data z certifikátů ukládají v různých formátech. Nejznámější formát PEM (Privacy Enhanced Mail) použitý pro rozšíření přenosu souborů přes e-mail, lze také použít formáty podle standardu PKCS#12 nebo DER formátu.

4.4.2 Certifikační autority

Certifikační autority vydávají certifikáty na základě ověření identity držitele veřejného klíče. Existuje několik stupňů ověření. Základní stupeň ověření, kdy je zaručeno pouze to, že v certifikační autorita nevydá dva různé certifikáty pro stejné jméno. Přes ověření, kdy se držitel musí prokázat platnými doklady, až po ověření nejen identity držitele platnými doklady a také ověřením, že má právo na operace ke kterým byl certifikát vydán.

Pro všechny uživatele nestačí jediná certifikační autorita, proto je zaveden strom certifikačních autorit. Kde certifikační autority může být certifikována jinými certifikačními autoritami. Základní model je stromová struktura, pro distribuci zátěže velkého počtu uživatelů. Všechny certifikační autority však nespadají pod jeden strom certifikačních autorit. Existují proto i křížové certifikace mezi stromy. To umožní komunikaci i uživatelům jejichž certifikáty vydali různé certifikační autority v různých stromech.



Obrázek 4.6: Stromy certifikačních autorit s křížovými certifikacemi [2]

5 Programovací platformy pro mobilní telefony

Všechny dnes běžně prodávané mobilní telefony mohou spustit uživatelskou aplikaci napsanou pro danou platformu daného zařízení. Programovací platformy se liší podle funkci, které zpřístupní na daném zařízení, tak podle velikosti výpočetního výkonu. Oficiální programovací platformy jsou podporovány přímo výrobci daných zařízení a existuje k nim tedy i dobrá podpora. Platformu vybíráme podle požadavků, které budou kladeny na aplikaci. Při výběru nelze také zapomenout na rozšířenost dané platformy na trhu.

5.1 JAVA ME

Java Platform, Micro Edition (neboli Java ME, dříve označovaná jako Java 2 Micro Edition nebo J2ME) je jedna ze tří platforem Javy (spolu s Java SE a Java EE). Tato platforma je vyvíjena společností Sun Microsystems. Takto sada virtuálního stroje a API umožňuje vytvářet a spouštět předkompilované skripty určené pro zařízení s malým výkonem, jako je mobilní telefon, PDA, apod. Díky univerzálnímu rozhraní jsou aplikace napsané pro platformu Java ME vysoce přenositelné a použitelné na zařízení od téměř jakéhokoliv výrobce. Tato programovací platforma se vyskytla mezi prvními. Dnes je podporována snad všemi mobilními telefony na trhu. Díky své univerzálnosti poskytuje však slabší výkon, a některé specifické funkce jsou podporovány pouze v rozšířených profilech API, nebo vůbec. [8][25]

5.2 Symbian OS

Symbian OS je proprietární operační systém, který byl navržen pro využití ve smartphonech. Slovem smartphone někdy také chytrý telefon se označuje telefon s otevřeným operačním systémem, který umožňuje využívat bohaté funkce přístroje a zároveň dovoluje přístroj rozšířit o mnoho dalších aplikací. Díky operačnímu systému mají aplikace mnohem širší pole působnosti. [18]

V roce 1998 se spojily firmy Psion, Nokia, Motorola a Ericsson, aby vytvořily sdružení Symbian, které prosazuje používání operačního systému EPOC. EPOC byl vyvinut firmou Psion. Po založení sdružení Symbian byl EPOC přejmenován na Symbian OS. Symbian je aktuálně vlastněn

společnostmi Nokia (největší podíl), Ericssom, Sony Ericsson, Panasonic, Siemens AG a Samsung. Dnes je Symbian jedničkou na trhu, používán především v telefonech Nokia.

Symbian OS je založen na mikrojádře, které obsahuje jen základní funkčnost pro běh operačního systému. Díky tomu je zlepšena robustnost, použitelnost a reakce na podměty od aplikací. V mikrojádře je obsažen pouze plánovač procesů, správa paměti a ovladače zařízení. Ostatní služby pro připojení do sítě, telefonování nebo souborový systém jsou umístěny ve vyšších vrstvách modelu operačního systému Symbian. Model Symbian OS obsahuje následující vrstvy:[5]

- vrstva rozhraní pro komunikaci s uživatelem (UI Framework Layer)
- vrstva aplikačních služeb (Application Services Layer)
- vrstva služeb operačního systému
- vrstva základních služeb

5.3 Windows Mobile

Windows Mobile je kompaktní operační systém kombinovaný se souborem základních aplikací pro mobilní zařízení, vyvinutý firmou Microsoft. Windows Mobile staví na Microsoft Win32 API. Podporované zařízení jsou smartphony, přenosné multimediální přehrávače nebo Pocket PC. Vzhledem se snaží přiblížit prostředí klasických Windows. Windows Mobile vyšlo v roce 2003 a vychází z Windows CE což je varianta operačního systému Windows určená pro vestavěná zařízení. Windows CE měl značně upravené jádro s podporou procesorů architektury Intel x86, MIPS, ARM a Hitachi SuperH. Windows Mobile jsou dnes druhou nejrozšířenější platformou na trhu se smartphony.

5.4 Ostatní

Na trhu v dnešní době existují i méně rozšířené programovací platformy. Jsou to buď platformy firem, které v nedávné době přišli na trh s mobilními telefony Apple nebo platformy různých skupin vyvíjející svobodné řešení. Platformy v této oblasti mají minimální zastoupení na trhu i podpora výrobců není tak rozšířená. V budoucnu se dá, ale čekat obrat hlavně v plně otevřených platformách.

5.4.1 iPhone

Společnost Apple v nedávné době uvedla na trh svůj první mobilní telefon s názvem iPhone. Tento telefon je v mnoha ohledech inovativní, hlavně v uživatelském ovládání. I přes ohromné množství prodaných kusů po jeho uvedení není zastoupení na trhu porovnatelné s platformami jako Symbina nebo Windows Mobile nijak vysoké. SDK bylo vydáno teprve v roce 2008.

5.4.2 Android

Společnost Google v nedávné době představila otevřenou platformu Android určenou pro mobilní telefony. Platforma Android obsahuje operační systém, mezivrstu oddělující aplikace od operáčního systému a soubor základních aplikací. Jako základní programovací jazyk byla zvolena Java, který vyvinula firma Sun Microsystems, pro svoji rozšířenost mezi vývojáři. Pro běh aplikací se však využívá virtuální stroj vyvinutý firmou Google. Tento provádí bytecode vytvořený překladačem pro Android platformu. Existuje však i program převodu bytecodu vytvořený původně standardním Java překladačem. Úplně základní běh zajišťuje Linuxové jádro verze 2.6. Pro platformu existuje množství knihoven. Platforma je zatím ve vývoji, fyzická zařízení běžící na této platformě by se měla začít prodávat koncem roku 2008. [19]

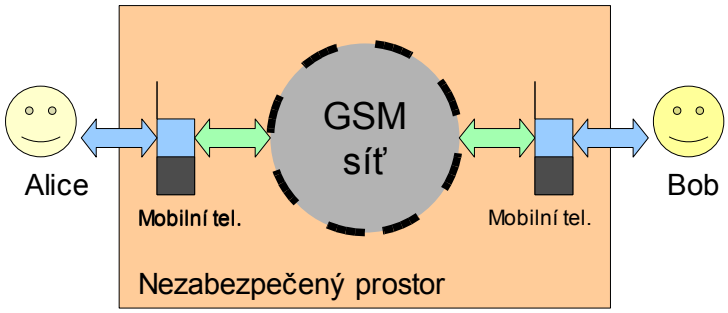
5.4.3 Linux

Existuje několik projektů portující operační systém Linux na určité typy mobilních telefonů. Paralelně se vyvíjí i několik projektů zabývající se uživatelským rozhraním a aplikační platformou pro mobilní zařízení. Takovýmto projektem je i Qtopia od společnosti Trolltech, která vyvinula grafickou knihovnu Qt. Qtopia dokáže být zatím spuštěna na mobilních telefonech Nokia n770, Nokia n880V, Fic Neo 1973 a Greephone. V lednu 2008 byla firma Trolltech koupena firmou Nokia. Můžete tedy v brzké době očekávat rozšíření Qtopie na více mobilních zařízení toho výrobce.

Zajímavým projektem je OpenMoko, který se snaží nejen připravit uživatelské prostředí pro mobilní telefony bežící na Linuxu, ale také postavit mobilní telefon s otevřenou specifikací hardwaru. První takovýto telefon nese název *Neo 1973* a jedná se o prototyp určený vývojářům. Projekt zaštiťuje společnost FIC.

6 Vývoj aplikace

Hlavní požadavek na aplikaci je zabezpečit důvěryhodnost informace poslané v SMS zprávě. Zabezpečení by mělo být dostatečně silné a mělo by mít vlastnosti moderního kryptografického systému. Na druhou stranu by šifrovací proces neměl příliš obtěžovat uživatele. Aplikace by neměla obsahovat zadní vrátka. Uživatelé se nemusejí nikdy fyzicky setkat a nemají k dispozici ani zabezpečený kanál pro distribuci šifrovacích klíčů.



Obrázek 6.1: Komunikace přes SMS zprávy

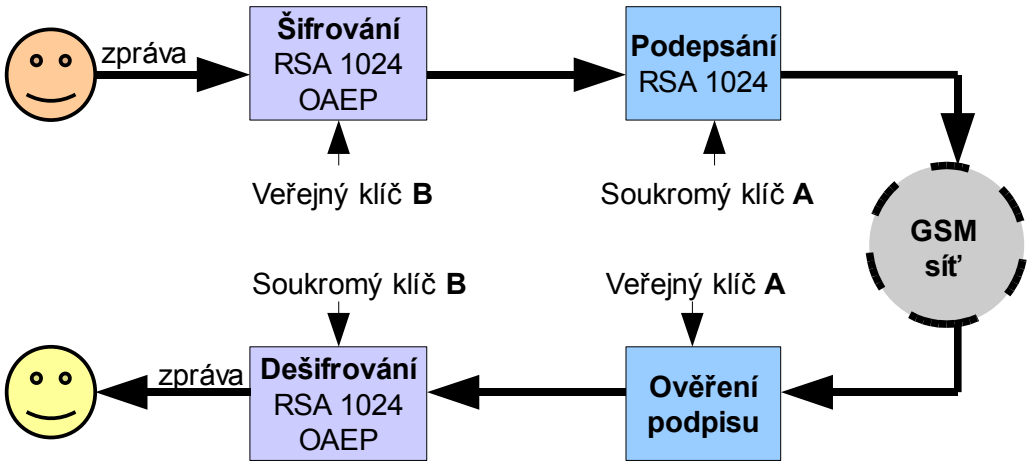
6.1 Návrh šifrovací metody

Pro zajištění důvěryhodnosti obsahu SMS zprávy byla zvolena asymetrická kryptografie. Šifrovací klíče uživatelů je možno distribuovat i přes méně bezpečné media. Uživatelé mají mnohem větší volnost než při použití symetrické kryptografie. Pro zamezení podvrhu veřejného klíče, jsou veřejné klíče uloženy v certifikátu. Certifikát lze použít jako žádost o podepsání certifikátu u certifikační autority. Certifikační autorita může daný certifikát s veřejným klíčem distribuovat, poskytuje-li tuto službu například prostřednictvím webového rozhraní. SMS zprávu lze nejen odposlechnou, ale i podvrhnout. Zde je využito další výhody asymetrické kryptografie – možnost digitálně podepisovat.

Na šifrovací algoritmus jsou následující požadavky:

- asymetrický
- umí šifrovat
- umí podepisovat
- moderní – bezpečnost závisí na síle klíče, ne na utajení algoritmu
- musí umět pracovat s výstupem menším než 1120 bitů (140 bytu – velikost SMS zprávy)
- neměl by zbytečně plýtvat místem
- bezpečný pro šifrování krátkého textu

Tyto požadavky splnil asymetrický algoritmus RSA, který lze použít jak k šifrování tak k digitálnímu podpisu dat. Velikost vstupních dat je rovno velikosti výstupních dat na rozdíl od algoritmu ElGamal. Velikost vstupu je u RSA rovna velikosti klíče. Klíč může mít hodnotu maximálně 1120 bitů, nejbližší nižší velikost standardního klíče je 1024 bitů. Tento šifrovací algoritmus však není příliš vhodný pro šifrování krátkého textu. Jedná se o deterministický algoritmu, který s délkou klíče 1024 bitů dokáže zašifrovat 128 znaků při 8 bitovém kódování. To ho činí velmi náchylný ke slovníkovým útokům. Tento problém lze vyřešit vhodným zarovnávacím schématem s nedeterministickou složkou. Dnes doporučovaným vyplňovacím schématem pro nové aplikace je OAEP. Díky OAEP je zamezeno slovníkovým útokům, ale je zkrácen prostor pro uživatelský text. Zašifrovaná zpráva je digitálně podepsána. Operace digitálního podpisu nepotřebuje využívat zarovnávacího schématu. Vstup je dlouhý přesně 1024 bitů. Pořadí šifrování a následně podpis bylo vybráno z důvodu omezených možností zarovnávacích schémat u digitálního podpisu. Zarovnávací schéma OAEP nebývá při podpisu implementováno a zarovnávání podle PKCS1 v1.5 při operacích se soukromým klíčem vyplňuje nevhodným způsobem. Při použití 1024 bitů z SMS zprávy zbývá ještě 96 bitů. Tento prostor bude použit pro vložení identifikátoru šifrované zprávy. Aplikace bude podle identifikátoru rozeznávat zabezpečené zprávy se kterými umí pracovat.



Obrázek 6.2: Popis šifrovací cesty

6.2 Výběr platformy a její vlastnosti

Z důvodu použití asymetrické kryptografie, která je náročnější na výpočetní výkon je nutné zvolit odpovídající programovací platformu. Pro svižný chod aplikace není příliš vhodná nejrozšířenější platforma Java ME [8]. Byla proto vybrána nejrozšířenější platforma pro smartphony Symbian OS. Aplikace byla vyvíjena pro verzi Symbian 9.1, které je kompatibilní také s nejnovějšími

telefony verze Symbian 9.2. Aplikace však není kompatibilní se staršími verzemi jako 8.1 a starší díky přepracovanému jádru řady 9.x [26].

Platforma Symbian je značně podporovaná hlavně ze strany firmy Nokia, která na svém webu zaštiťuje velkou řadu projektu pro tuto platformu. Vývojář si může vybrat z celé řady projektů doplněných o ilustrační příklady a bohatou dokumentaci. Pro tuto aplikaci byly vybrány projekty Open C, který posloužil jako kryptografický modul a projekt PyS60, který portuje programovací jazyk Python ve kterém byl napsán zbytek aplikace. [4]

6.2.1 Open C

Open C je technologie dovolující rychlejší vývoj aplikací pro platformu Symbian. Open C poskytuje známé C knihovny z Linuxových systémů. Dovoluje vývojářům použít již existující kód a zaměřit se na podstatné části jejich aplikace. Open C snižuje úsilí vynaložené na portování existujících komponent (například open source programů a desktopových aplikací) na platformu Symbian. Je zvláště vhodné pro části softwaru neobsluhující uživatelské rozhraní jako je jádro aplikace.

Knihovny obsažené v Open C jsou množina průmyslově standardizovaných POSIXových a midleware knihoven. Dobře známé API knihoven je nezměněno, to zvyšuje rychlost vývoje aplikace. Open C je navrženo pro maximální znovupoužití kódu. Některé z knihoven, které obsahuje Open C, jsou použity na projektech jako jsou Apache, Firefox nebo GMPlayer. Open C knihovny nejsou portovány celé, ale pokrývají více jak 75 procent funkcí obsažených ve standardních vydáních těchto knihoven. Open C obsahuje tyto knihovny:[20]

Tabulka 2: popis knihoven obsažených v Open C

Knihovna	**Popis**	**Open source projekt**	**Pokrývá [%]**
libc	Standardní C knihovna zahrnující standardní vstupně výstupní rutiny, databázové rutiny, bitové operace, řetězcové operace, znakové operace, DES šifrovací rutiny, funkce s časem a obsluhu signálů	OpenBSD (POSIX API)	47
libcrypt	Kryptografická knihovna obsahující funkce pro šifrování bloků dat, zpráv a hashování hesel	OpenSSL	100
libcrypto	Tyto knihovny poskytují OpenSSL implementaci Secure Socket Layer (SSL), Transport Layer Security (TLS) a S/MIME	OpenSSL	77
libdl	Poskytuje načítání DLL knihoven	POSIX	100

libglib	Knihovna pro rutinní účely. Např. mnoho použitelných datových typů, makra, typoke konverze, řetězcové utilyty atd. Pracuje na různých platformách Unix-like, Windows, OS/2 a BeOS	GNOME	100
libm	Aritmetické a matematické funkce	OpenBSD (POSIX API)	42
libpthread	Knihovna poskytuje IEEE std1003.1c (POSIX) standard pro implementaci vláken. Zahrnuje vytvoření zrušení vláken, rozhraní pro nastavení plánovače vláken, mutex a podmínkové proměnné pro synchronizaci přístupu ke sdíleným zdrojům.	OpenBSD (POSIX API)	60
libssl	Knihovna OpenSSL implementující protokolů SSL v2/v3 a TLS v1	OpenSSL	86
libz	Knihovna poskytuje funkce pro paměťovou kompresi a dekompresi zahrnující kontrolu integrity dat.	LIBZ	100
Celkově	Množství portovaných funkcí		78

6.2.2 PyS60

Projekt PyS60 dovoluje použít programovací jazyk Python na platformě Symbian OS. PyS60 umožňuje prototypování aplikací a hlavně jednoduchost při vytváření uživatelského rozhraní.

Python je obecně použitelný, vysokoúrovňový programovací jazyk. Jeho autor Guido van Rossum představil první verzi v roce 1991. Je navrhnut s důrazem na produktivitu programátora a čitelnost kódu. Python podporuje více programovacích paradigmat - primárně objektové, funkcionální a imperativní. Hlavní rysy jsou plně dynamické typování a automatická zpráva paměti.[6][21]

PyS60 zahrnuje všechny objektové typy a nejpoužívanější funkce standardní knihovny jazyka Python verze 2.2.2. PyS60 také zahrnuje řadu knihoven pro práci se samotným telefonem. Psaní uživatelského rozhraní v Pythonu je obecně velmi jednoduché. Programátor nemusí znát koncept stavby uživatelského rozhraní v C++ v systému Symbian. Modul pro tvorbu uživatelského rozhraní *appuifw* pro PyS60 je mnohem jednodušší. PyS60 je stále ve vývoji, proto nelze očekávat podporu všech možností, které poskytuje nativní API Symbianu v C++. Největším nedostatkem v PyS60 je nejspíše chybějící nativní podpora použití více jazyků pro lokalizaci aplikace. V porovnání efektivnosti na počet řádků kódu aplikace napsaná v Pythonu několikanásobně převyšuje nativní přístup v C++ [23].

Tabulka 3: Přehled modulů pro PyS60

Název modulu	**Poskytující funkce**
e32	Běžné služby Symbian OS
sysinfo	Přístup k systémovým informacím
appuifw	Framework uživatelského rozhraní
graphics	Služby související s vykreslováním grafických objektů
camera	Rozhraní pro fotografování s nahrávání videa fotoaparátem telefonu
keycapture	Rozhraní pro zachycení událostí vyvolaných klavesami
topwindow	Vytváření oken zobrazených před ostatními aplikacemi
gles	Mapování funkci OpenGL ES
glcanvas	Zobrazení grafických objektů OpenGL ES
sensor	Rozhraní pro práci se senzory telefonu
audio	Modul související s audio službami
telephone	Telefonní služby
messaging	Modul pro práci se zprávami
inbox	Přístup k uloženým zprávám
location	Informace o poloze v GSM síti
positioning	Rozhraní pro zjištění informací z GPS modulu
contacs	Práce s kontakty uloženými v telefonu
calendar	Přístup ke kalendáři
e32db	Rozhraní pro přístup k interní databázi v Symbian OS
e32dbm	Přístup k nativnímu API Database managment system (DBMS)
logs	Rozhraní pro přístup k logum

6.2.3 Možnosti uložení klíčů

Symbian OS využívá pro ukládání dat adresářovou strukturu, podobnou adresářové struktuře operačního systému Windows. Cesta k souboru nebo adresáři začíná jménem paměti (disku), následuje posloupnost adresářů a končí jménem souboru. V Symbian OS jsou tyto paměti:

- C: Paměť telefonu.
- D: Doplněk interní paměti telefonu.
- E: Paměťová karta. Paměťová karta je vyjímatelná a nemusí být vždy přítomna v telefonu.
- Z: Pevná paměť telefonu, do které nelze zapisovat.

Na úložiště veřejného a soukromého šifrovacího klíče jsou kladeny různé nároky, proto jsou klíče uloženy odděleně. Veřejný klíč by měl být rychle přístupný pro distribuci. Proto bylo vybráno umístnění v paměti telefonu, které je přístupné i vzdáleně. Zde je vytvořen adresář s názvem *cer*,

který slouží k uložení certifikátů s veřejnými klíči. Přesná cesta k tomuto umístnění je „C:\DATA\cer“. Veřejný klíč je umístěn v certifikátu standardu x509 verze 3, struktura dat uložených v souboru je podle standardu PEM.

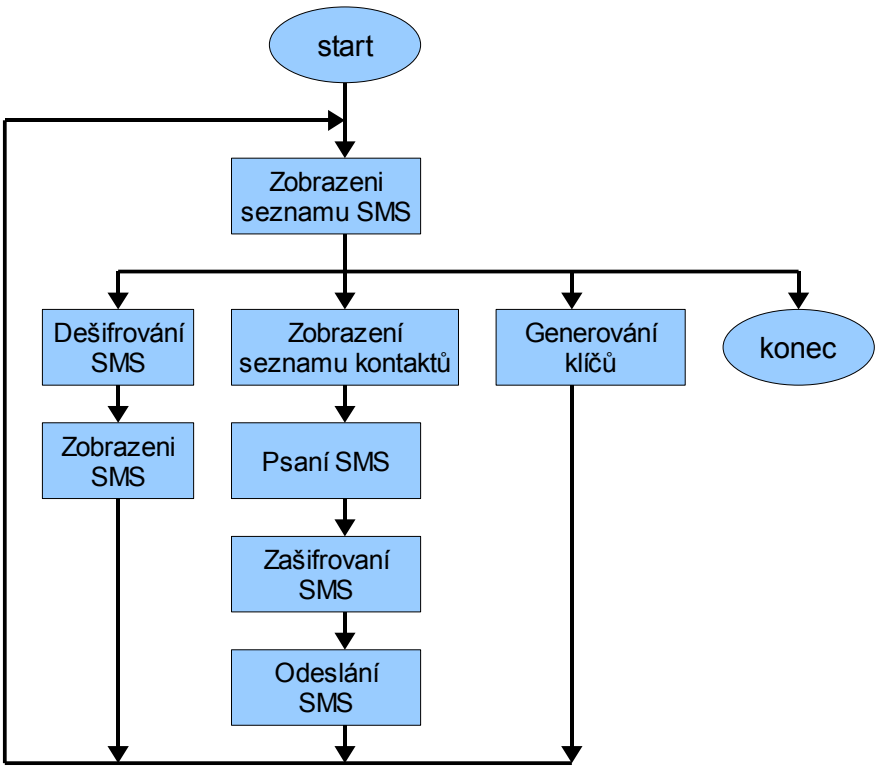
Soukromý klíč je tajný, nikde se nesmí distribuovat, proto je umístnění v privátním adresáři aplikace. Cesta k tomuto umístnění je: C:\Private\<vygenerované jméno>\ . Obsah adresáře není přístupná nikomu jinému než aplikaci, která je v něm nainstalovaná. Uživatel a ani útočník tedy nemá možnost s klíčem jakkoliv manipulovat. Klíč je uložen v souboru ve formátu dat PEM.

6.3 Implementace

6.3.1 Nároky na aplikaci

Aplikace pro šifrování SMS zpráv by měla mít tyto funkce:

- Uživatel může napsat, zašifrovat a poslat SMS zprávu příjemci, pokud zná jeho veřejný klíč a svůj soukromý klíč.
- Uživatel může dešifrovat SMS zprávu pokud zná veřejný klíč odesilatele a svůj soukromý klíč.
- Přijaté zašifrované SMS zprávy mohou být zobrazeny uživateli, bez nešifrovaných zpráv uložených v mobilním telefonu.
- Uživatel může obnovit seznam přijatých zašifrovaných SMS zpráv.
- Uživatel může generovat dvojice šifrovacích klíčů.



Obrázek 6.3: Návrh funkce programu

6.3.2 Stavba aplikace

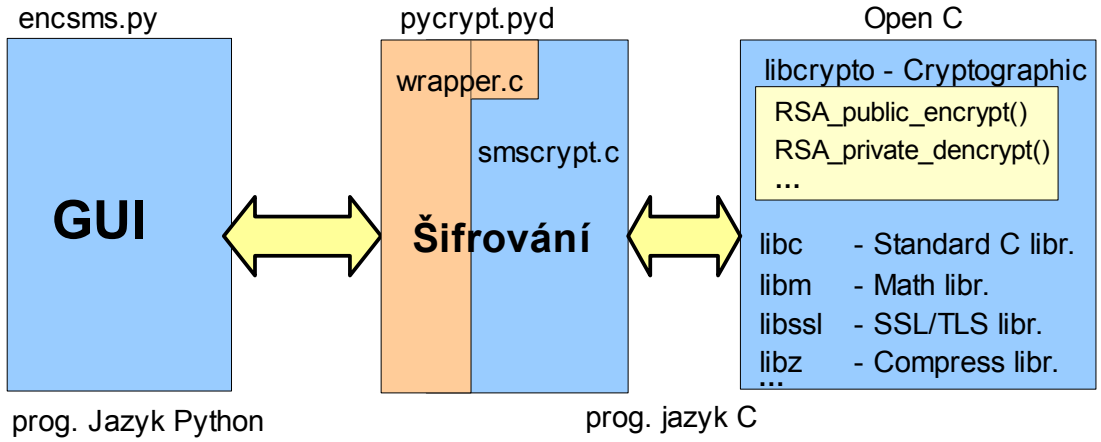
Aplikace se skládá z několika částí. Uživatelské rozhraní a práce s SMS zprávami je napsáno v jazyce Python. Šifrovací funkce jsou napsány v jazyce C, jsou využity knihovny z projektu Open C. Tyto dvě části bylo třeba spojit. Proto bylo implementováno také rozhraní pro komunikaci mezi těmito částmi. Toto schéma vychází z ukázkového příkladu [23], který poskytuje ukázku spojení programu v Pythonu a funkcí, které poskytuje Open C.

Grafické uživatelské rozhraní je reprezentováno třídami. Každá třída se stará o zobrazení dialogu k určité funkci a také aplikační logiku dané funkce. Implementované třídy jsou:

- *EncSMSApp* – základní třída, nastavení základních parametrů aplikace, kontrola prvního spuštění aplikace.
- *GenCert* – generování dvojice šifrovacích klíčů.
- *MsgView* – zobrazení obsahu zašifrované zprávy.
- *ReadView* – hlavní dialog se seznamem zašifrovaných SMS.
- *ContactsView* – dialog pro zobrazení certifikátů veřejných klíčů.
- *Message* – reprezentace zprávy, dešifrování zprávy.
- *SendView* – dialog pro psaní SMS zprávy, šifrování a odeslání zprávy.

Šifrování a ověření digitálního podpisu je implementováno v jazyce C společně s funkcemi pro konvertování volání mezi jazyky Python a C. Celá tato část je přeložena jako modul pro Python. Tento Modul se na začátky skriptu importuje jako jakýkoliv jiný běžný modul. Modul poskytuje tyto funkce:

- *encryptRSA* – zašifruje daný řetězec veřejným klíčem, který se nachází na daném umístnění.
- *decryptRSA* – rozšifruje daný řetězec soukromým klíčem, který se nachází na daném umístnění.
- *signRSA* – digitálně podepíše (zašifruje) soukromým klíčem, který se nachází na daném umístnění.
- *recoverRSA* – ověří digitální podpis (dešifruje) veřejným klíčem, který se nachází na daném .
- *generateRSA* – vygeneruje dvojici šifrovacích klíčů s danou délkou, veřejný klíč zapíše do certifikátu s danou platností a soubory uloží je na dané místa.
- *cert_info* – poskytne informaci o jménu (telefonním číslu), pro který byl vygenerován. Certifikát je načten z daného místa.



Obrázek 6.4: Struktura aplikace

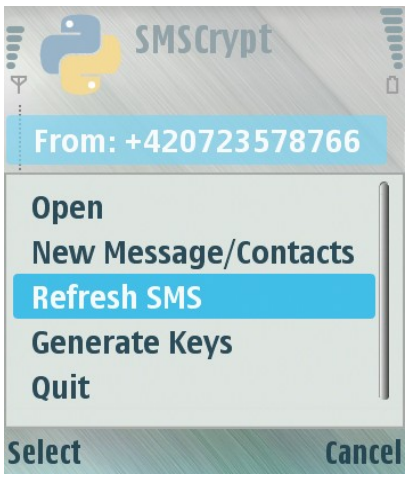
Obsah SMS zprávy je složen s identifikátoru a se zabezpečené 1024 bitové části. Identifikátor slouží pro rozpoznání šifrovaní SMS zprávy od běžné SMS zprávy. Identifikátor má tvar *#PyCr*. Následuje 1024 bitů dat. Data jsou kódována do 16 bitových unicode znaků, proto je celá zpráva kódována pomocí UCS-2. Při použití 8 bitového kódování SMS zprávy se vyskytly potíže.

6.3.3 Popis funkce aplikace

Po startu programu se kontroluje zda existuje adresář s certifikáty. Pokud neexistuje, znamená to že aplikace je spuštěna poprvé a je třeba vygenerovat šifrovací klíče. Generují-li se šifrovací klíče je uživatel upozorněn a požádán o zadání svého telefonního čísla. Tato informace je uložena na SIM kartě, ale API pro její zjištění není veřejné a lze ho použít jen skrze Symbian's partnership programs [22]. Dále je třeba zadat počet dní platnosti veřejného klíče, tato hodnota je společně s telefonním číslem vložena do certifikátu. Klíče jsou po úspěšném vygenerování uloženy na svá místa a je zobrazen hlavní dialog aplikace.

Hlavní dialog aplikace obsahuje seznam šifrovaných SMS zpráv. Pro výběr možných funkcí slouží levé tlačítko *Options*. Zde si může uživatel vybrat mezi funkcemi:

- *Open* – otevřít vybranou SMS zprávu.
- *New Message/Contacts* – přejde do nabídky se seznamem veřejných klíčů jiných uživatelům, kterým lze následně zaslat šifrovanou zprávu.
- *Refresh SMS* – obnoví seznam šifrovaných zpráv, vhodné zejména v případě, že přijde SMS zpráva za běhu aplikace.
- *Generate keys* – přejde ke generování nové dvojice šifrovacích klíčů
- *Quit* – ukončí aplikaci

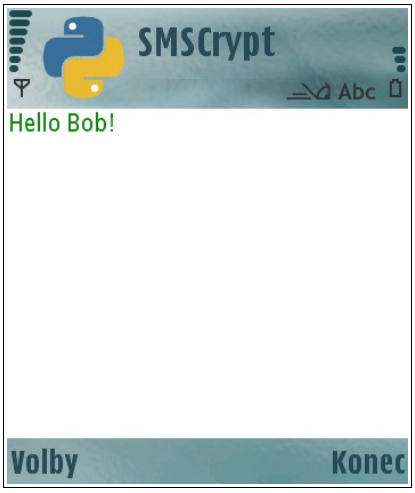


Obrázek 6.5: Hlavní dialog

Po vybrání položky *Open*, je načten obsah SMS zprávy a telefonní číslo odesilatele. Vyhledá se certifikát, který byl vydán pro toto telefonní číslo. Z certifikátu je načten veřejný klíč. Veřejným klíčem odesilatele načteným z certifikátu se ověří podpis a zpráva se dešifruje soukromým klíčem příjemce. Dešifrovaná zpráva je zobrazena uživateli.

Po vybrání položky *New Message/Contacts* je uživateli zobrazen seznam certifikátů uložených v adresáři *C:\DATA\cer* . Vybráním položky ze seznamu, se objeví dialog pro psaní nové zprávy.

Případně může certifikáty přejmenovat – položka *Rename* nebo zjistit ke kterému telefonnímu číslu se vážou – položka *Info*. Po napsání SMS zprávy je zobrazen dialog s telefonním číslem uloženým v certifikátu. Po jeho potvrzení je SMS zpráva zašifrována veřejným klíčem uloženým v certifikátu, podepsána soukromým klíčem a odeslána na uvedené číslo. Po odeslání se zobrazí hlavní dialog se seznamem šifrovaných zpráv.



Obrázek 6.6: Textový editor pro psaní SMS

6.4 Testování aplikace

Testování aplikace probíhalo na telefonech Nokia N80 a Nokia E65, každý měl vloženu SIM kartu od jiného Českého telefonního operátora. Testování ověřilo funkčnost řešení a poukázalo na vlastnosti, které by se daly v budoucnu zlepšit.

Nejkritičtější operací z výpočetního hlediska je generování klíčů. Byla testována samotná funkce pro generovaní klíčů *pycrypt.generateRSA* ve vytvořeném modulu. Testování probíhalo na telefonu Nokia N80, která má procesor architektury ARM9 běžící na 209 MHz. Pro test byla použita jednoduchá funkce, která načetla čas, provedla generování klíčů s danou délkou, znovu načetla čas a vypsala rozdíl časů. PyS60 dokáže pracovat pouze s přesností na jednotky sekund. Je vidět, že generování klíčů s velikostí 512 a 1024 bitů je rychlé, oproti generování klíčů s délkou 2048 bitů, zde již je generování kvalitního náhodného čísla problém. Generování klíčů délky 4096 bylo po několika minutách násilně ukončeno, zde byl již překročen časový interval, které by uživatel akceptoval.

Tabulka 4: Čas potřebný k vytvoření klíčů, testováno na telefonu Nokia N80

Velikost	**Měření č.1**	**Měření č.2**	**Měření č.3**	**Měření č.4**	**Měření č.5**	**Průměr**

klíčů [bity]	**[s]**	**[s]**	**[s]**	**[s]**	**[s]**	**[s]**
512	2	1	2	1	1	1,4
1024	6	4	6	5	5	5,2
2048	64	17	24	93	81	55,8

Testování správné funkčnosti OAEP vyplňování. Při šifrování zpráv stejného obsahu jsou výsledky šifrování různé. Zde se testovala samostatná funkce *encryptRSA* inmplementovaného modulu. Testovaní probíhalo šifrováním řetězce: „*testovací zprava*“ . Výsledky jsou následující, zapsané v 8bit kódování v Python řetězci:

- první zašifrování daného řetězce:

':r\xe8\xcco\x96\x90\xd2\xcb\x94\xb8\xe5\x85\xccA\xc1Yl\xb9\xd8PKpx$;o\xc9x\xae\
x08(Q\xb6b`\xd9E\x00\xea\xd8\x7f\xb0\x1e\xbe\xd8\x02&\x08B\xf0\xb8\x07\x01\xb9\x
f5\xc9\x1b\xe3\xc9\xd0\x9e\x10\x94 \xbft\xe2\x8a\xe7jF\x1esk\xff\xeb!W\xc9:\xdb\
x1f-\x02\xf8\xd5\x8aK^\x8c\xff\xcb\x1e\x1a\x13\xf0YB:\t7 wnEJ\xa6\xb3\xea\r\x9c\
xa7\xff\xfe\xda\xdaf\xb82\x7f\xcd\xb2\x08\xaa\x93\xd5^'

- druhé zašifrování daného řetězce:

'\xb5\xd3\xfd?=FmC\xbd*\x16\xf9\x9f\xc4\xe6\xb7m\xc2IP\x97I\x84nj<%\x02=%\x04\x1
5\x91\xf0\xdfuM\xc7\x83E\x14\xb9Q\x1a{>m\x19R\x9aWqb\xd5\xc5\xf0\xebiz\xc79)\r\x
9b>\xd2\xffI*$\x9d\xaf.\xd5\x96\x18\x19\xeb\x1aJ\x7f@\xf0\xc6\xbc\x18"?U\xc0\xea
\x90\xf4\xa55\xd0.\xe5\x99\x84\x9d\xa5s=\xd0\xec\xfc\x10\xb3R\xc8\x88\xc2r7~\xdc
A3\x19\x17\xc1\x15y]\x15\xc9Q'

Testování správné funkce generování klíčů a správné umístnění do certifikátu. Klíče byly vygenerovány v mobilním telefonu, následně pak zkopírovány do počítače a zde pomocí programu OpenSSL otevřeny. Při otevírání nebyla hlášena žádná chyba. Hodnoty klíčů se v jednotlivých souborech lišily, tedy každá vygenerovaná dvojice klíčů byla jiná.

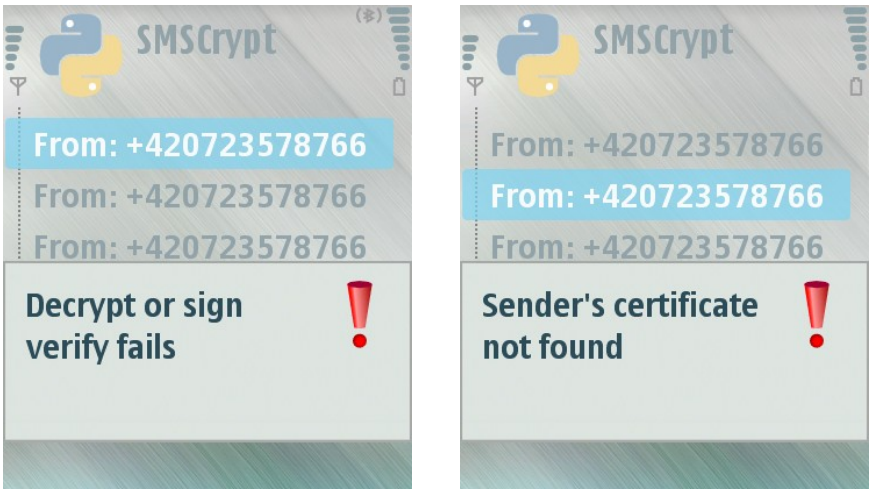
Při přeinstalování aplikace zůstane adresář s certifikáty na svém místě avšak soukromý klíč je smazán a je třeba vygenerovat druhý. Soukromý klíč nelze obnovit opětovnou instalací aplikace. Při přeinstalování aplikace zůstanou zachovány kontakty na ostatní uživatele, ale je jim třeba oznámit svůj nový veřejný klíč.

Při chybějícím nebo poškozeném souboru obsahující certifikát veřejného klíče nebo souboru obsahující soukromý klíč, je nahlášena chyba varovným upozorněním s textem „Sender's certificate not found“ nebo „Private Key opening fails“. Pro testování byl smazán soubor obsahující příslušný klíč.

Je-li SMS zpráva zašifrována nebo podepsána jiným klíčem než má příjemce je zobrazeno varovné upozornění o selhání dešifrovacího procesu „Decrypt or sign verify fails“. Pro testování byla poslána SMS zpráva zašifrovaná jedním klíčem, pak proběhlo generování nových klíčů a s těmito klíči se dešifrovalo.

Je-li přijata poškozená SMS zpráva je zobrazeno varovné upozornění o selhání dešifrovacího procesu. Poškození bylo simulováno odstraněním jednoho znaku v textu SMS zprávy.

Je-li poslána šifrovaná SMS zpráva na číslo ze kterého byla odeslána. Lze takovouto zprávu dešifrovat libovolnou dvojicí soukromého a veřejného klíče. Tato vlastnost je způsobena reverzibilitou šifrovacího algoritmu RSA. Při šifrování a podpisu odpovídajícím si párem klíčů, dojde k vyrušení operací. Zpráva, ale zůstane nečitelná z důvodu kódování dat do 16 bitových znaků a neodstranění OAEP zarovnání.



Obrázek 6.7: Chybová varování

7 Možnosti útoku, analýza rizik

Komunikace pomocí SMS zpráv není příliš bezpečná. Po zabezpečení SMS zpráv implementovanou aplikací se situace rapidně zlepší. Útočník musí na takto zabezpečenou komunikaci zvolit jiné typy útoků.

Hlavním cílem bylo zabránění odposlechu SMS zprávy. SMS zpráva se dá odposlechnout v různých částech sítě GSM. Odposlech při odesílání zprávy z mobilního zařízení k dané BTS není příliš vhodný. SMS zpráva je šifrována pomocí algoritmu A5. Existují útoky na tento algoritmus, ale vyžadují větší objem odposlechnutých informací. Odposlech musí být fyzicky realizován v relativní blízkosti mobilního zařízení, z důvodu nízkého vysílacího výkonu. V případě odposlechu na straně příjemce je situace obdobná. Uživatel však může být ve větší vzdálenosti, stačí mu být v dosahu správné BTS.

Komunikace je šifrovaná jen mezi mobilním zařízením a BTS. Má-li útočník fyzický přístup k BTS může odposlouchávat všechny spojení probíhající skrze tuto stanici. SMS zprávy musí také projít SMS centrem, které se postará o jejich správné doručení. Zde mohou být uloženy i delší dobu, dokud nedojde k jejich doručení nebo jim nevyprší doba platnosti. Obvykle má každý operátor jedno SMS centrum. Útočník s přístupem k SMS centru může odposlouchávat všechny SMS zprávy přenášené v síti daného operátora.

Měl-li útočník fyzický přístup k mobilnímu telefonu, mohl provést klonování SIM karty. Pak se může do sítě přihlásit jako oprávněný uživatel a SMS zprávy budou doručeny přímo mu. Na trhu se také objevily aplikace pro odposlech. Aplikace je vytvořena pro platformu Symbian OS. Po instalaci přeposílá aplikace na zvolené telefonní číslo všechny příchozí a odchozí SMS. Dokáže také upozornit na příchozí a odchozí hovory odesláním SMS zprávy na zvolené telefonní číslo, případně dovolují odposlech formou konferenčního hovoru – útočník zavolá na dané číslo a hovor je přepnut na konferenční mezi třemi účastníky. Aplikace může být odinstalována po doručení SMS ve speciálním tvaru a uživatel tedy nemusí nikdy zjistit, že byl odposloucháván [27].

Proti těmto útokům je zabezpečení SMS zprávy šifrováním vhodné. V případě, že útočník odposlechne SMS zprávu musí ji před zneužitím dešifrovat. Dešifrování RSA šifry o délce klíče 1024 bitů nebylo dosud úspěšně realizováno. Nejlepší výsledek je zatím z roku 2005, kde Jens Franke z Univerzity v Bonnu dokázal faktorizovat číslo o délce 663 bitů [2].

Útočník musí pro dešifrování znát soukromý klíč příjemce. Soukromý klíč příjemce se nikde nepřesunuje a je stále uložen v paměti telefonu. Toto umístnění je přístupné pouze šifrovací aplikaci nainstalované v telefonu, jiné aplikace ani uživatelé zde nemají přístup. Existuje však způsob jak tuto složku zpřístupnit. Do telefonu se musí nainstalovat aplikace, která zprostředkuje vypnutí kontroly

přístupu do chráněných adresářů. Pro vypnutí kontroly přístupu musí být telefon připojen k počítači. Zpřístupnění není časově náročné a útočník může lehce získat soukromý klíč uživatele. Kontrola přístupu se znovu aktivuje po restartu telefonu. [27]

Útočník se může také pokusit podvrhnout veřejný klíč příjemce uložený u odesilatele. Při úspěšném pokusu, budou doručovány SMS zprávy na útočníkovo telefonní číslo. Doručené SMS zprávy bude schopen také dešifrovat. Útočník však již není chopen zprávu podepsat jako správný odesilatel a přeposlat ji příjemci.

Při podvrhnutí veřejného klíče u příjemce, může útočník předstírat oprávněného odesilatele. V případě, že se útočníkovi podaří podvrhnout veřejné klíče na obou stranách, může odposlouchávat, upravovat a přeposílat veškerou komunikaci prostřednictvím zašifrovaných SMS zpráv. Tyto útoky však nejsou ničím novým v asymetrické kryptografii, proto jsou použity pro zabezpečení veřejných klíčů certifikáty.

Útočník však nemusí pouze odposlouchávat SMS zprávy, ale může také zkusit SMS zprávy podvrhnout. Při tomto útoku je třeba odeslat SMS zprávu, která má správné číslo odesilatele. Toho lze dosáhnout napadnutím SMS centra, nebo napadnutí komunikačního kanálu mezi SMS centrem a SMS branou. Z toho SMS centra je pak možno odeslat SMS zprávu s libovolným číslem odesilatele. Službu odesílání SMS zpráv s libovolným číslem odesilatele si lze dnes i komerčně předplatit, u několika internetových firem. Útočník sice dokáže odeslat zašifrovanou zprávu příjemci, ale nedokáže ji korektně podepsat.

Útočník může také využít možnosti odstranění podpisu od zašifrované zprávy. Zprávu musí odposlechnout, pak odstranit podpis původního odesilatele a podepsat ji svým soukromým klíčem. Takto připravenou zprávu může přeposlat původnímu příjemci. Útočník musí znát kontext komunikace mezi odesilatelem a příjemcem, aby docílil správné reakce příjemce. Aplikace je primárně určena pro komunikaci mezi lidmi, proto se nepředpokládá širší uplatnění tohoto útoku. V případě, že by komunikace probíhala mezi stroji, které by přenášely data nebo příkazy byla by situace vážnější.

8 Možnosti rozšíření

Při testování a analýze možných útoků byly zjištěny vlastnosti aplikace, které by se mohli v budoucnu vylepšit nebo pozměnit. Jsou zde také uvedeny vlastnosti, které jsou implementovány v komerčních produktech, a mohly by být využity v této aplikaci.

Přístup do aplikace přes heslo. Tato vlastnost by zabránila případnému útočníku číst SMS zprávy v po krátkou dobu, kdy není mobilní telefon střežen oprávněným uživatelem. Bude-li mít útočník dlouhodobější přístup k telefonu nebo telefon dokonce odcizí, není přístup k aplikaci přes heslo překážkou.

Ukládání klíčů v šifrované podobě. V případě, že by došlo k odcizení mobilního telefonu, musel by útočník soukromí klíč nejen zkopírovat z paměti telefonu, ale také dešifrovat.

Aplikace by mohla umožňovat lepší práci s kontakty v telefonu. Přiřazení jména uloženého pod číslem v telefonních kontaktech k veřejnému klíči. Tato vlastnost by pomohla lepší orientaci při komunikaci s ostatními uživateli.

Aplikace by mohla umět zabezpečit také MMS zprávy. MMS zprávy mají mnohem větší kapacitu přenosu. Bylo by možné tedy využít aplikaci pro bezpečnou výměnu dat.

Aplikace by mohla ověřovat certifikáty on-line na požádání. Bylo by třeba připojení k síti internet a ověření platnosti certifikátu u certifikační autority nejspíše pomocí Online Certificate Status Protocol (OCSP). Aplikace by také mohla stahovat a odesílat certifikáty s veřejnými klíči z Certifikační autority.

Aplikace by mohla kontrolovat u přijatých zpráv číslo SMS centra. Tímto by se upozornilo na přijetí podezřelé SMS zprávy. V případě že by se útočník dostal k soukromému klíči odesilatele pak může zprávy i správně podepisovat a skrze nezabezpečené SMS centrum zasílat SMS zprávy i s odpovídajícím telefonním číslem odesilatele.

9 Závěr

Tato diplomová práce se zabývá zabezpečením SMS zpráv v mobilní komunikaci. V první části je popsána funkce a struktura mobilní sítě. Jsou zde popsány jednotlivé subsystémy, které zajišťují komunikaci mezi účastníky. Dále jsou také popsány vlastnosti, kódování a přenosu SMS zpráv v mobilní síti. Následuje kapitola s popisem moderních šifrovacích algoritmů vhodných pro zabezpečení SMS zpráv.

Před implementací bylo třeba navrhnout metodu pro šifrování SMS zpráv a vhodný způsob výměny šifrovacích klíčů. Pro šifrování byla zvolena asymetrická kryptografie. Na proces výměny šifrovacích klíčů se nekladou tak přísné požadavky jako při použití symetrické kryptografie. Pro zabezpečení SMS zpráv využívá algoritmu RSA. Využívá jak operace šifrování tak operace digitálního podpisu. Tímto je SMS zabezpečena nejenom proti odposlechu, ale také proti podvrhnutí. Tato část byla vytvořena v rámci semestrálního projektu a byly tak splněny první dva body zadání, seznámit se s přenosem SMS zpráv a návrhem šifrovací metody.

Dále bylo třeba vybrat vhodnou programovací platformu pro mobilní telefony. V kapitole jsou srovnány jednotlivé platformy podle vlastností a rozšíření. Byla vybrána nejrozšířenější platforma pro smartphony Symbian OS. V další kapitole jsou popsány podpůrné projekty, které byly využity pro implementaci aplikace. Následují implementační detaily. Zde je popsáno propojení komponent, ze kterých se aplikace skládá a výsledná funkce aplikace. Implementací aplikace byl splněn třetí bod zadání.

Implementovaná aplikace byla podrobena řadě testů. Testy byly zaměřeny jak na ošetření chybových stavů, tak i na ověření správné funkce zabezpečení SMS zpráv. Testováno bylo na mobilních telefonech Nokia N80 a Nokia E65.

V předposlední kapitole jsou diskutovány možné útoky jak na běžnou SMS komunikaci, tak na zabezpečenou SMS komunikaci pomocí této aplikace. V poslední kapitole jsou podle výsledků testů, analýzy možných útoků a srovnáním s komerčními aplikacemi navrhnuty vylepšení. Tímto byl splněn i poslední bod zadání.

Byly splněny všechny body zadání diplomové práce. Hlavní přínos této práce je v popsání vlastností a implementace aplikace využívající k zabezpečení asymetrickou kryptografii. Dostupné bakalářské a diplomové práce se zatím zabývaly zabezpečením SMS zpráv pomocí symetrické kryptografie.

Literatura

[1] Hanus, S.: Rádiové a mobilní komunikace, elektronická skripta, Fakulta elektrotechniky a komunikačních technologií, Brno, 2002, s. 83

[2] Hanáček, P.: Přednášky k předmětu Kryptografie. c2008 [citováno 2008-01-03] Dostupný z WWW: https://www.fit.vutbr.cz/study/courses/KRY/private/

[3] Hanáček P., Staudek J.: Bezpečnost informačních systémů. ÚSIS, Praha, 2000, s. 127, ISBN 80-238-5400-3.

[4] Fitzek, F., Reichert, F.: Mobile Phone Programming and its Application to Wireless Networking, Springer 2007, s. 473, ISBN 978-1-4020-5969-8

[5] Harrison, R.: Programujeme aplikace Symbian OS v jazyce C++ / Vyd. 1., Brno : Computer Press, 2006, s. 407, ISBN 80-251-1243-8

[6] Harms, D.: Začínáme programovat v jazyce Python / 1. vyd. Brno : Computer Press, 2003. xviii, 458 s. ISBN 80-7226-799-X

[7] Szturc, J.: Šifrátory pro mobilní telefony, bakalářská práce, Brno: Fakulta elektrotechniky a komunikačních technologií, 2007

[8] Žitovský, O.: Šifrování komunikace mobilních zařízení pomocí Java ME, diplomová práce, Fakulta informatiky, Masarykova universita, Brno 2007

[9] Kaliski, B.: PKCS #1: RSA Encryption Version 1.5, RFC2313. Network Working Group, 1998, c2008 [citováno 2008-05-03], Dostupný z WWW: http://tools.ietf.org/html/rfc2313

[10] Wikipedie: Otevřená encyklopedie: Global System for Mobile Communications. c2008 [citováno 2008-01-03], Dostupný z WWW: http://cs.wikipedia.org/wiki/Global_System_for_Mobile_Communications

[11] Web Tomáše Richtra: Základní struktura síťě GSM. c2008 [citováno 2008-01-03] Dostupný z WWW: http://tomas.richtr.cz/mobil/gsm-strukt.htm

[12] Wikipedie: Otevřená encyklopedie: Caesarova šifra. c2008 [citováno 2008-01-03], Dostupný z WWW: http://cs.wikipedia.org/wiki/Caesarova_%C5%A1ifra

[13] Wikipedie: Otevřená encyklopedie: Symetrická kryptografie. c2008 [citováno 2008-01-03], Dostupný z WWW: http://cs.wikipedia.org/wiki/Symetrick%C3%A1_kryptografie

[14] Wikipedie: Otevřená encyklopedie: Asymetrická kryptografie. c2008 [citováno 2008-01-03], Dostupný z WWW: http://cs.wikipedia.org/wiki/Kryptografie_s_ve%C5%99ejn%C3%BDm_kl%C3%AD%C4%8Dem

[15] Wikipedia: : The Free Encyklopedia: Short mesage service. c2008 [citováno 2008-01-03], Dostupný z WWW: http://en.wikipedia.org/wiki/Short_message_service

[16] Internetový server Mobilmania: Už vím proč má SMS jen 160 znaků. c2008 [citováno 2008-01-03], Dostupný z WWW: http://www.mobilmania.cz/default.aspx?article=1107583

[17] Internetový server společnosti Computer Press a.s.: Velký průvodce protokoly TCP/IP: bezpečnost. c2008 [citováno 2008-01-03], Dostupný z WWW: http://www.cpress.cz/knihy/tcp-ip-bezp/CD-0x/3.html

[18] Wikipedie: Otevřená encyklopedie: Smartphone. c2007 [citováno 2008-01-03], Dostupný z WWW: http://cs.wikipedia.org/wiki/Smartphone

[19] Internetový server Google: Android – An Open Handset Projekt. c2008 [citováno 2008-05-03], Dostupný z WWW: http://code.google.com/android/index.html

[20] Internetový server nokia.com: Open C. C2008 [citováno 2008-02-04], Dostupný z WWW: http://sw.nokia.com/id/8271db20-cf52-4bb9-91ad-5ff5338f485f/Open_C_Developer_Productivity_v1_2_en.pdf

[21] Wikipedia: : The Free Encyklopedia: Python, c2008, [citováno 2008-01-03], Dostupný z WWW: http://en.wikipedia.org/wiki/Python_%28programming_language%29

[22] Internetový server wiki.forum.nokia.com: How to get my own phone number, c2008, [citováno 2008-01-03], Dostupný z WWW: http://wiki.forum.nokia.com/index.php/How_to_get_my_own_phone_number

[23] Internetový server www.forum.nokia.com: SMS crypto example, c2008, [citováno 2008-01-03], Dostupný z WWW: http://www.forum.nokia.com/info/sw.nokia.com/id/79410fde-246d-4731-9d10-3e2c5fdde953/Open_C_SMS_Crypto_Example_v1_1_en.zip.html

[24] Wikipedia: : The Free Encyklopedia: OAEP, c2008, [citováno 2008-01-03], Dostupný z WWW: http://en.wikipedia.org/wiki/Optimal_Asymmetric_Encryption_Padding

[25] Wikipedie: Otevřená encyklopedie: Java ME. c2008, [citováno 2008-01-03], Dostupný z WWW: http://cs.wikipedia.org/wiki/Java_ME

[26] Wikipedie: Otevřená encyklopedie: Symbian OS. c2008, [citováno 2008-01-03], Dostupný z WWW: http://en.wikipedia.org/wiki/Symbian_OS

[29] Internetový server Symbianforum: Návod pro zpřístupnění složek C:\Private a C:\SYS c2008, [citováno 2008-01-03], Dostupný z WWW: http://www.symbianforum.cz/viewtopic.php?t=46734&rnd=218294&sid=6795599cae8a51cdfb2066255d404a41

[30] Inertetový denník idnes: Vyzkoušeli jsme špionský mobil, vaší oběti vezme veškeré soukromí, c2008 [citováno 2008-01-03], Dostupný z WWW: http://mobil.idnes.cz/vyzkouseli-jsme-spionsky-mobil-vasi-obeti-vezme-veskere-soukromi-109-/mob_tech.asp?c=A080227_164709_mob_tech_vok

Příloha A: Obsah přiloženého CD

Na přiloženém CD jsou přiloženy tyto soubory:

- Zdrojové kódy aplikace.
- Text diplomové práce ve formátech pdf a odt.
- Uživatelská příručka s popisem instalace a nastavení všech částí nezbytných pro přeložení aplikace.
- Instalační soubory aplikace a knihoven do mobilního telefonu.
- SDK pro mobilní platformu Symbian OS v9.1 – S60 3rd Edition.
- Vývojové prostředí Carbide C++ v1.2 .
- Instalační balík Open C obsahující plug-in do SDK, instalační soubory do mobilního telefonu a základní dokumentaci.
- PyS60 instalační soubor do mobilního telefonu, instalační soubor obsahující grafický interpret s možností Bluetooth připojení k PC a plug-in PyS60 do SDK.
- Adresář s elektronickými knihami a manuály ze kterých bylo čerpáno.
- OpenSSL instalační soubor do PC, zdrojové kódy
- ensymble_python2.5-0.26.py skript pro překlad Python skriptů do instalačních balíčků pro mobilní telefony
- Python pro PC verze 2.5
- Perl verze 5.6.1 – nutný pro běh SDK
- Java Runtime – nutná pro běh SDK